Modelli del Mondo, JEPA e Intelligenza Incarnata

Introduzione

La tesi, il filo e l'architettura dell'argomentazione

I. La domanda fondamentale

Che cosa significa, per un sistema artificiale, *comprendere il mondo*? La domanda non è retorica né puramente filosofica: ha una risposta tecnica precisa, o almeno una direzione verso cui una risposta può essere cercata, e l'intera struttura argomentativa di questa monografia ruota attorno a essa. La comprensione, nell'accezione in cui il termine è qui adoperato, non coincide con la capacità di descrivere il mondo — di etichettare oggetti in un'immagine, di tradurre una frase, di rispondere a una domanda. Essa consiste piuttosto nella capacità di *anticipare* il modo in cui il mondo si trasformerà in seguito a eventi o azioni, e di utilizzare questa anticipazione per pianificare, adattarsi e agire in modo coerente con obiettivi a lungo termine. Un sistema che comprende il mondo nel senso qui definito dispone di ciò che la letteratura chiama un *modello del mondo* — una rappresentazione interna della struttura causale e dinamica dell'ambiente — e sa impiegarlo per simulare mentalmente le conseguenze delle proprie scelte prima di attuarle.

La tesi centrale di questa monografia può essere formulata in modo diretto: fra tutte le architetture attualmente proposte per apprendere modelli del mondo da dati sensoriali grezzi, la *Joint Embedding Predictive Architecture* gerarchica (JEPA gerarchica), nella formalizzazione proposta da Yann LeCun e sviluppata nelle sue varianti visive e temporali, offre le fondamenta teoriche più solide e le prospettive empiriche più promettenti. Non si afferma che essa sia già una soluzione compiuta — i capitoli conclusivi di questa trattazione sono dedicati proprio ai problemi aperti che ne limitano la portata attuale — ma che la direzione da essa tracciata sia quella corretta, e che le alternative oggi dominanti portino in sé difetti strutturali che nessuna quantità di dati aggiuntivi o di potenza computazionale può colmare.

Questa introduzione ha un duplice scopo: esplicitare il filo argomentativo che attraversa i ventiquattro capitoli dell'opera, e giustificare perché ciascuno di essi sia un passo *necessario* di una dimostrazione, non un'escursione laterale né una semplice aggiunta enciclopedica.

II. Il problema delle rappresentazioni e l'errore dei modelli generativi

La storia degli approcci ai modelli del mondo nell'apprendimento automatico è, nella sua fase moderna, dominata da un'intuizione di fondo apparentemente ragionevole: per simulare come il mondo evolverà, occorre imparare a *riprodurlo*. Se un sistema è capace di generare predizioni fedeli dei propri input sensoriali futuri — di ricostruire il fotogramma successivo di un video, di prevedere la prossima osservazione di un agente in un ambiente fisico — allora ha necessariamente catturato qualcosa di essenziale sulla struttura dinamica del mondo. Questa intuizione è alla radice della famiglia degli *autoencoder variazionali* applicati alle sequenze, del *Recurrent State Space Model* di Hafner e collaboratori, e più in generale dell'intera linea di ricerca che culmina in DreamerV3, il sistema di reinforcement learning model-based più capace attualmente disponibile.

L'obiezione di LeCun, formulata nel contributo seminale del 2022 e ricostruita analiticamente nel terzo capitolo di questa trattazione, è che questa intuizione, sebbene plausibile in superficie, è fondamentalmente errata. Un modello generativo che predice le osservazioni future — i pixel di un fotogramma, i campioni di un segnale audio, le coordinate giunzionali di un robot — deve distribuire la propria capacità rappresentativa su un'enorme quantità di dettagli irrilevanti per qualsiasi obiettivo di pianificazione. La texture di una superficie, il pattern di illuminazione su un oggetto, il rumore di fondo di un segnale: tutto ciò deve essere modellato con la stessa attenzione accordata alle strutture dinamiche che hanno rilevanza causale per l'azione. Il costo computazionale è enorme; la difficoltà di addestramento è corrispondentemente elevata; e — aspetto più sottile e più fondamentale — il paesaggio di ottimizzazione tende a essere dominato dalla componente di ricostruzione dei dettagli irrilevanti, piuttosto che dall'apprendimento delle invarianti strutturali che rendono utile il modello.

La conseguenza operativa di questo difetto è il fenomeno dell'*errore di compounding*, analizzato in forma rigorosa nel secondo capitolo: pianificando per rollout di lunghezza H nel modello appreso, ogni errore di predizione si propaga e si amplifica, poiché traiettorie che si discostano anche lievemente dalla distribuzione di addestramento producono stati per i quali il modello non ha mai ricevuto supervisione. Quando il modello opera in spazio delle osservazioni, questo fenomeno è particolarmente grave, perché lo spazio delle osservazioni è ad alta dimensione e la manifold delle osservazioni plausibili è una sottovarietà di misura nulla rispetto allo spazio ambiente.

III. La soluzione JEPA: predire rappresentazioni, non osservazioni

La risposta proposta dalla JEPA è concettualmente semplice e matematicamente elegante. Invece di addestrare il modello a predire le osservazioni future nello spazio originale — i pixel, le coordinate — si addestra a predire le *rappresentazioni* di quelle osservazioni in uno spazio latente appreso congiuntamente. La predizione avviene interamente nello spazio latente: dato il contesto codificato s_x = f_θ(x_ctx), il predittore h_φ produce ŝ_y = h_φ(s_x, pos_y), e la loss è una distanza nello spazio delle rappresentazioni tra ŝ_y e la rappresentazione target sg(g_ψ(x_y)), dove sg(·) indica l'operazione di stop-gradient e g_ψ è un encoder target aggiornato come media mobile esponenziale di f_θ.

Questa formulazione risolve il problema della predizione irrilevante: lo spazio latente è appreso proprio in modo da catturare le strutture che sono predittivamente rilevanti, scartando i dettagli che non lo sono. Non viene esercitata alcuna pressione a ricostruire l'input originale; la sola pressione è quella di formare uno spazio in cui le relazioni di prossimità riflettano la struttura causale e temporale del mondo. In termini di teoria dell'informazione, la JEPA implementa implicitamente il principio dell'*information bottleneck* di Tishby, Pereira e Bialek: massimizzare l'informazione mutua tra rappresentazione e variabile target, preservando la quantità minima necessaria dall'input originale.

Il pericolo simmetrico è il *collasso della rappresentazione*: se il predittore è abbastanza espressivo, può imparare a ignorare l'input e predire una costante, o a collassare tutte le rappresentazioni in un punto, azzerando la loss senza apprendere alcunché di utile. Le tecniche anti-collasso — stop-gradient, momentum encoder nella tradizione di Bootstrap Your Own Latent, regolarizzazione della covarianza nella tradizione di VICReg, e la loro analisi come soluzioni al problema dell'equilibrio tra collasso e uniformità — costituiscono uno dei nuclei tecnici del terzo capitolo e sono centrali per comprendere perché la JEPA funzioni.

La *JEPA gerarchica*, nella concezione di LeCun, estende questo principio a più livelli di astrazione. A ogni livello l della gerarchia, un encoder f^(l)_θ mappa le rappresentazioni del livello inferiore in rappresentazioni di livello superiore, più astratte e invarianti; un predittore h^(l)_φ opera su queste rappresentazioni di livello superiore per anticipare stati futuri su orizzonti temporali più lunghi; le rappresentazioni di livello inferiore sono necessarie per predire il dettaglio fine, quelle di livello superiore per pianificare su orizzonti estesi. Questo schema corrisponde, non accidentalmente, alla struttura gerarchica del sistema visivo dei mammiferi, dove le aree corticali di livello superiore codificano concetti sempre più astratti e invarianti rispetto alle trasformazioni sensoriali.

IV. L'architettura dell'argomentazione

La struttura dell'opera rispecchia il percorso logico dal problema alla soluzione, dalla soluzione alle sue fondamenta, dalle fondamenta alle sue applicazioni e ai suoi limiti.

Il *primo capitolo* stabilisce la genealogia concettuale del modello del mondo: dai *small-scale models of reality* di Kenneth Craik (1943) attraverso la cibernetica di Norbert Wiener, le mappe cognitive di Edward Tolman, il filtro di Kálmán come primo modello del mondo formalmente preciso, fino alle architetture neurali contemporanee. Questo capitolo non è un ornamento storico: è necessario perché la tesi della JEPA gerarchica acquista tutto il suo peso solo se si comprende che essa risponde a un problema — quello di costruire rappresentazioni interne dell'ambiente che siano allo stesso tempo compatte, predittivamente rilevanti e gerarchicamente organizzate — che la tradizione del controllo ottimo, della psicologia cognitiva e della cibernetica aveva già identificato nei suoi tratti essenziali, pur senza disporre degli strumenti per risolverlo.

Il *secondo capitolo* costruisce il linguaggio matematico nel quale tutti i successivi risultati sono formulati. La derivazione rigorosa dell'Evidence Lower Bound (ELBO), la struttura del Recurrent State Space Model (RSSM) con la sua separazione tra stato deterministico h_t e stato stocastico z_t, il training della policy nell'ambiente immaginario tramite gradiente analitico attraverso il rollout — tutto ciò è sviluppato con attenzione alle condizioni di stabilità numerica e alle fonti di bias nell'addestramento. Questo capitolo è necessario perché senza padronanza del modello generativo di riferimento — Dreamer nella sua versione più matura — non è possibile comprendere i difetti strutturali che la JEPA si propone di correggere.

Il *terzo capitolo* introduce la JEPA: il manifesto di LeCun come critica ai modelli generativi, la formalizzazione dei modelli a base energetica come cornice teorica unificante, l'architettura dettagliata di I-JEPA e V-JEPA, le strategie anti-collasso, e infine la visione gerarchica completa. È il capitolo pivot dell'intera monografia: prima di esso si costruisce il problema; dopo di esso si esplora la soluzione nelle sue ramificazioni.

Il *quarto capitolo* porta la JEPA nell'ambito della robotica incarnata, dove il modello del mondo è chiamato non solo a comprendere ma ad agire: dinamica differenziabile del robot, model predictive control nello spazio latente, trasferimento sim-to-real, politiche visuomotorie. La tesi di questo capitolo è che la JEPA si presta naturalmente all'integrazione con il controllo model-based, poiché lo spazio latente in cui opera il predittore è sufficientemente strutturato da permettere l'ottimizzazione delle traiettorie di azione, a differenza di quanto accade con gli spazio di pixel dove la curvatura del paesaggio di costo rende l'ottimizzazione intrattabile.

I *capitoli dal quinto al tredicesimo* costituiscono ciò che si potrebbe definire l'infrastruttura teorica della tesi. Ciascuno affronta una dimensione del problema che non è accessoria ma strutturale.

Il quinto capitolo è dedicato all'apprendimento auto-supervisionato: i metodi contrastivi come istanza del principio di massimizzazione dell'informazione mutua, il collasso dimensionale e la dimensione intrinseca delle rappresentazioni, il Contrastive Predictive Coding come antenato formale della JEPA, la relazione tra distillazione della conoscenza e auto-supervisione implicita. Questo capitolo è necessario perché la

JEPA è fondamentalmente un metodo di self-supervised learning, e comprenderne le proprietà richiede padronanza del paesaggio teorico in cui si inserisce.

Il sesto capitolo tratta la formulazione in tempo continuo dei modelli del mondo: le Neural ODE come limite continuo dei modelli di transizione discreti, la derivazione del metodo aggiunto per il calcolo efficiente del gradiente, le reti hamiltoniane e lagrangiane come architetture che incorporano la struttura fisica della conservazione dell'energia, l'integrazione simplettica per preservare la struttura geometrica del flusso simplettico nello spazio delle fasi. La necessità di questo capitolo deriva dalla constatazione che i sistemi fisici — robot, veicoli autonomi, oggetti in un ambiente — non evolvono a passi discreti ma continuamente, e che imporre la struttura della meccanica hamiltoniana al modello del mondo riduce drasticamente lo spazio delle ipotesi, migliorando la capacità di generalizzazione e la robustezza dell'apprendimento.

Il settimo capitolo formalizza la pianificazione sotto incertezza attraverso il framework dei POMDP, la rappresentazione per iperpiani della funzione di valore, la distinzione rigorosa tra incertezza epistemica e aletoristica, e il principio di energia libera variazionale di Friston come unificazione di percezione e azione sotto un unico criterio di ottimizzazione. Questo capitolo stabilisce che un modello del mondo, per quanto accurato, deve essere integrato in un framework decisionale che gestisca esplicitamente l'incertezza; la JEPA non è immune da questa necessità, e la sua integrazione con l'inferenza attiva e con i metodi di esplorazione basati sulla curiosità è uno dei problemi aperti discussi nel capitolo conclusivo.

L'ottavo capitolo esplora le leggi di scala e le capacità emergenti: come la quantità di parametri, di dati e di calcolo determina qualitatively le proprietà dei modelli appresi, e cosa significa che un modello del mondo fondazionale — addestrato su un corpus vastissimo e poi adattato a compiti specifici — potrebbe essere la giusta architettura per l'intelligenza artificiale generale. Genie, UniSim, SORA come casi studio di modelli del mondo fondazionali; il teorema di approssimazione universale per le sequenze come fondamento teorico della capacità di questi modelli. La connessione con la tesi della JEPA gerarchica è diretta: se le leggi di scala funzionano, e se la JEPA è l'architettura giusta, allora la crescita della scala dovrebbe produrre emergenze qualitative in modo più efficiente che non con i modelli generativi tradizionali.

Il nono capitolo introduce la geometria differenziale riemanniana e l'information geometry come strumenti per analizzare la struttura degli spazi latenti. La metrica di Fisher come tensore riemanniano sulle famiglie statistiche, il gradiente naturale come ottimizzatore che rispetta la geometria dell'informazione, le varietà riemanniane apprese come pullback metrics dalla rete neurale, la geometria iperbolica per le rappresentazioni gerarchiche. Questo capitolo è necessario perché la qualità di uno spazio latente non è solo una questione statistica ma geometrica: uno spazio latente ben strutturato per un modello del mondo dovrebbe avere una metrica tale che le traiettorie dinamicamente simili siano vicine, e le transizioni causalmente differenti siano lontane. La JEPA, residendo nello spazio latente, dipende in modo critico dalla qualità geometrica di questo spazio.

Il decimo capitolo analizza il paesaggio di loss dell'ottimizzazione profonda: la connettività tra minimi (*mode connectivity*), la sharpness-aware minimization come tecnica per trovare minimi piatti con migliore generalizzazione, le barriere di energia tra bacini di attrazione. La motivazione è duplice: da un lato, comprendere perché l'addestramento della JEPA converge a soluzioni di buona qualità richiede di capire la struttura del paesaggio di ottimizzazione; dall'altro, la robustezza del modello del mondo rispetto a perturbazioni degli input dipende dalla piattezza del minimo in cui la rete risiede.

L'undicesimo capitolo è dedicato al meta-learning: come un sistema che ha appreso molti modelli del mondo di compiti diversi può adattarsi rapidamente a nuovi compiti con pochi esempi. MAML e le sue varianti come metodo per apprendere un'inizializzazione dei parametri che sia vicina al minimo per qualsiasi compito della distribuzione; le reti neurali iperboliche e i modelli latenti condizionali come alternative. La connessione con la JEPA gerarchica è profonda: un sistema che ha appreso a costruire rappresentazioni gerarchiche del mondo dovrebbe essere in grado di adattare la parte alta della gerarchia a nuovi compiti mantenendo fissa la parte bassa, in analogia con il transfer learning nei sistemi visivi biologici.

Il dodicesimo capitolo introduce l'apprendimento causale come componente necessaria di qualsiasi modello del mondo che aspiri a una generalizzazione genuina. Le reti causali e i modelli strutturali, i test di indipendenza condizionale come fondamento dell'identificazione causale, la differenza tra correlazione e causalità nel contesto dell'apprendimento delle rappresentazioni. La tesi di questo capitolo è che un modello del mondo appreso puramente da correlazioni statistiche — anche un modello molto accurato entro la distribuzione di addestramento — sarà sistematicamente fuorviato in distribuzione, perché le correlazioni statistiche cambiano al variare del regime, mentre le relazioni causali sono invarianti sotto interventi. La JEPA gerarchica, operando su rappresentazioni astratte, si avvicina implicitamente a questo obiettivo ma non lo raggiunge per costruzione; l'integrazione esplicita di vincoli causali è una delle direzioni aperte più promettenti.

Il tredicesimo capitolo formalizza la teoria dell'attenzione e dei Transformer come substrato computazionale dei modelli del mondo contemporanei. La derivazione completa del meccanismo di self-attention da principi primi, le proprietà teoriche di approssimazione, FlashAttention come ottimizzazione IO-efficiente, sparse attention e long-context architectures. La necessità di questo capitolo è pratica oltre che teorica: tutte le architetture di rilievo — I-JEPA, V-JEPA, Dreamer nelle sue versioni più recenti, i modelli del mondo fondazionali — impiegano Transformer come backbone, e comprenderne le proprietà è indispensabile per valutare le scelte architetturali.

I *capitoli dal quattordicesimo al diciottesimo* estendono la tesi centrale a domini più ampi.

Il quattordicesimo capitolo ricollega il modello del mondo agli algoritmi di reinforcement learning model-based: Dyna come schema concettuale fondamentale, MuZero come istanza dell'utilizzo del modello del mondo per la pianificazione MCTS in spazio latente, le proprietà teoriche di convergenza degli algoritmi model-based rispetto a quelli model-free. La tesi argomentata è che la JEPA, fornendo uno spazio

latente di qualità superiore, migliora tutte le componenti di questi algoritmi: la qualità della simulazione immaginaria, la densità informativa dei rollout, la stabilità del training della policy.

Il quindicesimo capitolo affronta la questione dei modelli di linguaggio come modelli del mondo impliciti: in che misura un Large Language Model che ha appreso a prevedere il prossimo token su un corpus vastissimo ha appreso anche un modello del mondo fisico e sociale? Le evidenze empiriche suggeriscono capacità sorprendenti di simulazione mentale e ragionamento causale, ma anche difetti sistematici riconducibili alla natura puramente simbolica e non incarnata dell'apprendimento. La connessione con la JEPA gerarchica è qui biforcuta: da un lato, i LLM potrebbero fungere da componente simbolica di alto livello in un'architettura JEPA gerarchica multimodale; dall'altro, la JEPA potrebbe fornire ai modelli di linguaggio le rappresentazioni grounded nel mondo fisico di cui attualmente difettano.

Il sedicesimo capitolo esplora le architetture di memoria — episodica, di lavoro, a lungo termine — come estensione del modello del mondo oltre l'orizzonte di un singolo episodio. Le reti con memoria differenziabile, i modelli a recupero neurale, l'apprendimento continuo come problema di aggiornamento del modello del mondo in assenza di catastrophic forgetting. La necessità di questo capitolo emerge dalla constatazione che un sistema dotato di intelligenza genuina non può affidarsi solo alla memoria implicita nei pesi della rete: deve poter memorizzare e recuperare esperienze specifiche, aggiornare il proprio modello del mondo in risposta a eventi nuovi, e farlo senza degradare la conoscenza precedentemente acquisita.

Il diciassettesimo capitolo tratta i modelli generativi avanzati — flussi normalizzanti, mappe di trasporto ottimale, modelli a score — come modelli del mondo impliciti. Questi formalismi, pur appartenendo a una tradizione teorica diversa dalla JEPA, convergono verso di essa nel momento in cui si considerano le loro varianti che operano in spazio latente anziché in spazio delle osservazioni. La relazione formale tra trasporto ottimale, divergenza di Wasserstein e training di modelli generativi è sviluppata in forma rigorosa, con derivazioni complete.

Il diciottesimo capitolo affronta il grounding multimodale: come la JEPA può essere estesa a input di diverse modalità — visione, linguaggio, audio, dati tattili — e come queste modalità possono essere allineate in uno spazio latente condiviso senza perdere le strutture specifiche di ciascuna. Le architetture visio-linguistiche come CLIP e le loro varianti come istanze di JEPA multimodale, la questione della *semantic completeness* della rappresentazione latente.

I *capitoli dal diciannovesimo al ventunesimo* situano la tesi centrale nelle cornici teoriche più ampie della fisica statistica e della topologia.

Il diciannovesimo capitolo applica gli strumenti della fisica statistica all'apprendimento profondo: le transizioni di fase nel training dei modelli del mondo, la replica theory come fondamento della teoria della generalizzazione, i fenomeni di condensazione nelle rappresentazioni latenti al crescere del dataset. Questo non è un capitolo speculativo: è un programma di ricerca che ha già prodotto risultati formali sulla

relazione tra architettura, dimensione del dataset e qualità della rappresentazione appresa.

Il ventesimo capitolo impiega gli strumenti della topologia algebrica — l'analisi persistente, i diagrammi di persistenza, i numeri di Betti come invarianti topologici — per caratterizzare la struttura geometrica degli spazi latenti appresi dalla JEPA. La *topological data analysis* applicata ai modelli del mondo è un campo emergente che promette di fornire garanzie formali sulla qualità delle rappresentazioni indipendentemente dalle specifiche di architettura e addestramento.

Il ventunesimo capitolo sintetizza le tensioni e le convergenze teoriche: dove la JEPA gerarchica e le teorie dell'inferenza attiva, della causalità, della geometria riemanniana e delle leggi di scala convergono verso una visione coerente, e dove invece rimangono incompatibilità che richiedono ulteriore sviluppo.

I *capitoli dal ventiduesimo al ventiquattresimo* affrontano i problemi aperti e i requisiti pratici.

Il ventiduesimo capitolo mappa le frontiere aperte: i limiti attuali della JEPA gerarchica, le connessioni con la neuroscienza computazionale, le direzioni di ricerca più promettenti.

Il ventitreesimo capitolo — uno dei due aggiunti a correzione del difetto strutturale originario della monografia — affronta l'apprendimento per rinforzo model-based *offline*: come addestrare politiche sicure ed efficaci a partire da dataset fissi di traiettorie, senza interazione con l'ambiente. MOPO, MOReL, MOBILE come istanze di questo paradigma; la teoria della generalizzazione conservativa come fondamento formale per il trade-off tra sfruttamento del modello e conservatorismo rispetto alla distribuzione dei dati; la sinergia specifica tra rappresentazioni JEPA e MBRL offline come direzione aperta di rilievo.

Il ventiquattresimo capitolo tratta la robustezza certificata e la sicurezza formale come requisito non opzionale per il dispiegamento di sistemi autonomi basati su modelli del mondo. La predizione conforme applicata ai sistemi di controllo, l'analisi di raggiungibilità Hamilton-Jacobi, le funzioni barriera di controllo, la sintesi di shield e la logica temporale lineare come strumenti per garantire la sicurezza in senso formale. La connessione con la JEPA è diretta: le proprietà formali dello spazio latente — la sua regolarità, la controllabilità della dinamica latente, la possibilità di verificare proprietà sulle rappresentazioni — determinano se e in che misura le garanzie formali di sicurezza possono essere trasferite dall'ambiente simulato all'ambiente reale.

V. Una nota sullo stile formale

L'opera adopera sistematicamente la notazione descritta nell'Appendice A. Si segnalano le scelte non ovvie: le variabili latenti deterministiche sono indicate con h, quelle stocastiche con z, in conformità con la tradizione RSSM di Hafner; i parametri

di encoder, decoder e predittore sono indicati con θ, φ, ψ rispettivamente; l'operazione di stop-gradient è indicata con $\mathrm{sg}(\cdot)$ e il prodotto di Hadamard con $\odot$. Tutte le derivazioni matematiche sono riportate per esteso nella posizione in cui compaiono per la prima volta, senza rimandi ad appendici per i passaggi intermedi; le appendici contengono esclusivamente materiale supplementare che non interrompe il filo argomentativo principale.

Le dimostrazioni formali vengono distinte dalle argomentazioni euristiche mediante segnalazione esplicita; si usa il simbolo $\square$ per concludere una dimostrazione e la locuzione *si può mostrare che* per indicare risultati il cui sviluppo completo richiederebbe uno spazio sproporzionato rispetto alla loro importanza nel contesto. Questa scelta risponde a un'esigenza di onestà intellettuale: un'opera che pretendesse di derivare tutto *ex nihilo* sarebbe non solo inutilmente prolissa, ma potenzialmente fuorviante sulla solidità relativa dei vari risultati.

VI. Il valore della tesi e il suo perimetro

È opportuno precisare che cosa questa monografia *non* sostiene. Non si sostiene che la JEPA gerarchica sia già un'architettura completa per l'intelligenza artificiale generale: le sue implementazioni attuali — I-JEPA per la visione statica, V-JEPA per il video — non sono ancora sistemi capaci di pianificazione, di interazione incarnata o di ragionamento astratto. Non si sostiene che le architetture generative — Dreamer, i modelli di diffusione, i flussi normalizzanti — siano prive di valore: esse hanno prodotto risultati empirici di assoluto rilievo e continuano a essere strumenti potenti in numerosi domini. Non si sostiene che l'apprendimento per rinforzo model-free sia obsoleto: in molti regimi pratici esso rimane competitivo o superiore.

Ciò che si sostiene è più preciso e più modesto, ma teoricamente più profondo: che i *principi architetturali* della JEPA — predizione nello spazio latente, apprendimento congiunto di encoder e predittore, gerarchia di rappresentazioni a scale temporali multiple — siano i principi corretti per costruire sistemi che comprendono il mondo nel senso formulato all'apertura di questa introduzione. E che la convergenza di argomenti provenienti dalla teoria dell'informazione, dalla geometria differenziale, dalla fisica statistica, dalla neuroscienza computazionale e dalla pratica ingegneristica, tutti convergenti verso questo stesso insieme di principi, costituisca un'evidenza indiziaria di rilievo a sostegno di questa tesi, anche in assenza di una dimostrazione deduttiva definitiva.

È su questo piano — non quello della competizione tra sistemi nei benchmark, ma quello dei principi che guidano la progettazione — che questa monografia intende offrire un contributo duraturo.

I. Fondamenti concettuali e storici dei modelli del mondo

1. La nozione di modello interno nella tradizione cibernetica e cognitiva

L'idea che un sistema intelligente debba disporre di una rappresentazione interna dell'ambiente in cui opera non è un'invenzione recente dell'apprendimento automatico. Essa affonda le radici in una tradizione intellettuale che attraversa la cibernetica degli anni Quaranta, la psicologia cognitiva del dopoguerra e la teoria del controllo ottimo sviluppata nel corso degli anni Sessanta e Settanta. Comprendere questa genealogia è indispensabile per apprezzare la profondità teorica delle architetture odierne e per non ridurle a semplici artifizi ingegneristici.

Il fisiologo e filosofo Kenneth Craik fu tra i primi, nel 1943, a formulare in termini espliciti la tesi che il cervello costruisce "modelli su scala ridotta della realtà" (*small-scale models of reality*) che lo mettono in condizione di ragionare sulle conseguenze di azioni future senza doverle effettivamente eseguire. Nell'opera *The Nature of Explanation*, Craik sostiene che la mente traduce eventi esterni in simboli, opera su questi simboli seguendo relazioni analoghe a quelle del mondo fisico, e ritraduce il risultato in azioni od osservazioni. Questa visione triangolare — percezione, elaborazione interna, azione — costituisce il prototipo concettuale di ogni successiva formalizzazione del modello del mondo.

Negli stessi anni, Norbert Wiener fondava la cibernetica come scienza del controllo e della comunicazione negli animali e nelle macchine. Il concetto di retroazione negativa (*negative feedback*) implicava già una forma rudimentale di modello interno: il regolatore confronta uno stato desiderato con quello corrente e agisce per minimizzarne la discrepanza. Tuttavia, la cibernetica classica si limitava a modelli reattivi, incapaci di anticipare il futuro oltre l'istante immediatamente successivo. La svolta verso la pianificazione e la previsione a lungo orizzonte richiedeva strumenti matematici più ricchi.

La psicologia sperimentale fornì un contributo decisivo con gli esperimenti di Edward Tolman sulle mappe cognitive dei ratti. Tolman dimostrò che i roditori in labirinti complessi apprendono non semplici catene di risposte riflesse, bensì strutture spaziali coerenti che permettono loro di tracciare nuovi percorsi verso la meta dopo che il percorso abituale è stato bloccato. Questa capacità di ricombinare la conoscenza acquisita in modo flessibile suggeriva l'esistenza di una rappresentazione interna dell'ambiente con proprietà topologiche non banali. Tolman coniò il termine *cognitive map* nel 1948, anticipando di decenni la nozione informatica di grafo di stato.

La teoria del controllo ottimo, e in particolare il lavoro di Rudolf Kálmán nel 1960 sui filtri che portano il suo nome, rese formalmente precisa l'idea di stima ricorsiva dello stato di un sistema dinamico. Il filtro di Kálmán può essere interpretato come un

meccanismo che mantiene una distribuzione gaussiana sullo stato latente del mondo e la aggiorna in modo bayesianamente ottimale a ogni nuova osservazione. Si tratta, nella sostanza, di un modello del mondo lineare-gaussiano con inferenza esatta. La rappresentazione probabilistica dello stato latente, e il dualismo tra un modello di transizione (che descrive come il mondo evolve) e un modello di osservazione (che descrive come le percezioni dipendono dallo stato), sono esattamente le componenti che ritroviamo, enormemente generalizzate, nelle architetture neurali contemporanee.

2. Verso una definizione formale: stato latente, transizione, osservazione

Prima di addentrarsi nelle architetture specifiche, è opportuno fissare una nomenclatura rigorosa. Si consideri un agente che interagisce sequenzialmente con un ambiente: all'istante discreto t l'agente riceve un'osservazione o_t appartenente a uno spazio di osservazione $\mathcal{O}$, compie un'azione a_t appartenente a uno spazio di azioni $\mathcal{A}$, e riceve uno scalare di rinforzo r_t. L'ambiente possiede uno stato interno s_t appartenente a uno spazio di stati $\mathcal{S}$, generalmente non direttamente accessibile all'agente (ambiente parzialmente osservabile).

Un modello del mondo è una coppia di distribuzioni di probabilità parametriche:

p_θ(s_t | s_{t-1}, a_{t-1}) — modello di transizione
p_φ(o_t | s_t) — modello di osservazione

a cui si aggiunge, per i compiti con rinforzo, un modello di ricompensa:

p_ψ(r_t | s_t)

Il modello di transizione codifica la dinamica del mondo: dato lo stato al passo precedente e l'azione eseguita, descrive la distribuzione di probabilità sullo stato corrente. Il modello di osservazione codifica come le percezioni sensoriali dipendono dallo stato del mondo. Quando θ, φ e ψ sono reti neurali profonde, si parla di *deep world model*.

La stima dello stato latente corrente a partire dalla storia delle osservazioni e delle azioni si chiama *inferenza dello stato* e corrisponde al calcolo della distribuzione posteriore:

q(s_t | o_{1:t}, a_{1:t-1})

Nei modelli lineari-gaussiani questa distribuzione è anch'essa gaussiana e il filtro di Kálmán ne calcola ricorsivamente la media e la covarianza. Per modelli non lineari e non gaussiani, l'inferenza esatta è computazionalmente intrattabile, e si ricorre a

tecniche approssimate: filtro di Kálmán esteso, filtri particellari, o, nel caso delle reti neurali profonde, inferenza variazionale.

3. L'era moderna: il contributo seminale di Ha e Schmidhuber

L'articolo "World Models" di David Ha e Jürgen Schmidhuber, pubblicato nel 2018, segna l'ingresso dei modelli del mondo nell'agenda dell'apprendimento per rinforzo profondo in modo che potremmo definire spettacolare. L'architettura proposta si articola in tre componenti distinte ma cooperanti, la cui composizione ricorda — non senza intenzionalità da parte degli autori — l'organizzazione neurale del cervello dei mammiferi.

Il primo modulo è un autoencoder variazionale (VAE) che comprime ogni fotogramma osservato o_t in un vettore di variabili latenti z_t di dimensione bassa (tipicamente 32 o 64 scalari). Questo modulo svolge una funzione analoga alla corteccia visiva primaria: estrarre una rappresentazione compatta e astratta degli input sensoriali.

Il secondo modulo è una rete ricorrente con mixture density output (MDN-RNN), che apprende la dinamica temporale nello spazio latente. Ricevendo in ingresso la concatenazione (z_t, a_t), la rete produce una distribuzione di probabilità sulla variabile latente successiva z_{t+1}, parametrizzata come miscela di gaussiane:

$$p(z_{t+1} \mid z_t, a_t, h_t) = \Sigma_{k=1}^{K} \pi_k(h_t) \cdot \mathcal{N}(z_{t+1};\ \mu_k(h_t),\ \sigma_k^2(h_t))$$

dove h_t è lo stato nascosto della rete ricorrente (un LSTM nel lavoro originale), e π_k, μ_k, σ_k sono funzioni parametriche dello stato nascosto. La scelta di una miscela di gaussiane, anziché una singola gaussiana, è motivata dalla natura intrinsecamente multimodale dell'incertezza sul futuro: molte conseguenze di un'azione sono ugualmente plausibili a priori.

Il terzo modulo è un controllore lineare estremamente semplice: un singolo strato densamente connesso che mappa la concatenazione (z_t, h_t) in a_t:

$$a_t = W_c \cdot [z_t; h_t] + b_c$$

I parametri W_c e b_c vengono ottimizzati con la strategia evolutiva CMA-ES (Covariance Matrix Adaptation Evolution Strategy), che evita il calcolo del gradiente e risulta particolarmente efficiente per spazi dei parametri piccoli come quello del controllore lineare.

La trovata più originale del lavoro consiste nel fatto che il controllore può essere addestrato interamente nell'ambiente simulato dalla MDN-RNN — il cosiddetto

addestramento "nel sogno" (*training in the dream*). L'agente genera traiettorie immaginarie campionando dallo stato nascosto della rete ricorrente, ne valuta le prestazioni attraverso un segnale di rinforzo approssimato, e aggiorna i parametri del controllore di conseguenza. Il trasferimento al mondo reale (in questo caso all'emulatore del videogioco Doom) richiede solo un aggiustamento fine.

Questa separazione netta tra un sistema percettivo-predittivo (VAE + MDN-RNN) e un sistema decisionale (controllore) riflette una scelta architetturale deliberata: la capacità di apprendere modelli accurati del mondo dovrebbe, secondo gli autori, essere separata dalla capacità di pianificare e decidere. Questa intuizione riemerge, con una formalizzazione molto più raffinata, nelle architetture successive.

4. Il problema dell'accuratezza del modello e il compounding error

Un'obiezione fondamentale ai modelli del mondo riguarda il fenomeno noto come *errore di compounding* (*compounding error* o *drifting*). Quando si pianifica nel mondo simulato rollando il modello di transizione per H passi nel futuro, l'errore di predizione si accumula: se ad ogni passo il modello introduce un'imprecisione ε in norma L2, dopo H passi l'errore totale può crescere come $O(H \cdot \varepsilon)$ nel caso ottimistico lineare, ma tipicamente molto più rapidamente quando le dinamiche sono caotiche o quando le traiettorie divergono dal regime di addestramento.

Formalmente, sia s_t lo stato reale al tempo t e $\hat{s}_t$ lo stato stimato dal modello. La divergenza si può caratterizzare attraverso la derivata di Lyapunov della dinamica errata. Si definisce l'errore di stato $e_t = s_t - \hat{s}_t$; la sua evoluzione è governata da:

$$e_{t+1} = f(s_t, a_t) - \hat{f}(\hat{s}_t, a_t)$$
$$\approx \partial f/\partial s \cdot e_t + [f(s_t, a_t) - \hat{f}(s_t, a_t)]$$

dove il primo termine rappresenta l'amplificazione dell'errore corrente attraverso il jacobiano della dinamica reale, e il secondo termine rappresenta l'errore sistematico del modello allo stato reale. Se il raggio spettrale di $\partial f/\partial s$ supera l'unità — condizione comune per sistemi instabili o caotici — l'errore cresce esponenzialmente.

Le strategie per mitigare questo problema includono: l'utilizzo di orizzonti di pianificazione brevi, l'addestramento del modello sulla distribuzione degli stati visitati dalla politica corrente (distribuzione *on-policy*), l'incertezza consapevole attraverso modelli ensembled o bayesiani, e l'apprendimento di rappresentazioni latenti che rendano l'errore di predizione più controllabile. Quest'ultima strategia è al cuore della JEPA, che discuteremo estesamente nella Parte III.

5. La prospettiva bayesiana: modelli come prior sul mondo

Una lettura alternativa e complementare dei modelli del mondo è quella offerta dalla teoria bayesiana della percezione, riassunta nel principio del cervello predittivo di Karl Friston e nella più ampia tradizione dell'*inferenza attiva*. In questa prospettiva, il cervello non è un semplice rilevatore di stimoli, bensì un sistema generativo che costruisce continuamente previsioni sugli input sensoriali attesi e aggiorna le proprie rappresentazioni in proporzione allo scarto — l'*errore di predizione* — tra ciò che attende e ciò che effettivamente riceve.

Il modello generativo del cervello definisce una distribuzione congiunta su stati latenti s e osservazioni o:

$$p(o, s) = p(o \mid s) \cdot p(s)$$

La percezione corrisponde all'inferenza della distribuzione posteriore $p(s \mid o)$, mentre l'azione viene selezionata in modo da minimizzare l'energia libera variazionale F, che funge da limite superiore alla sorpresa:

$$\begin{aligned} F[q, o] &= \mathbb{E}_q[\log q(s) - \log p(o, s)] \\ &= KL[q(s) \,||\, p(s \mid o)] - \log p(o) \\ &\geq -\log p(o) \end{aligned}$$

dove q(s) è una distribuzione approssimante variazionale. Minimizzare F rispetto ai parametri della distribuzione approssimante q equivale a ottimizzare l'inferenza percettiva; minimizzare F rispetto alle azioni — modificando le osservazioni future attese — costituisce invece l'azione nel senso della teoria di Friston.

Questa formulazione elegante unifica percezione e azione sotto un unico principio variazionale e suggerisce che un agente razionale dovrebbe sempre cercare di ridurre la propria incertezza sul mondo — un impulso epistemico che va al di là della semplice massimizzazione della ricompensa strumentale. La connessione con i modelli del mondo neurali è profonda: le reti di transizione apprendono implicitamente un modello generativo dello spazio degli stati, e la divergenza KL nella loss di addestramento variazionale può essere interpretata come una forma di regolarizzazione dell'energia libera.

6. Rappresentazione, astrazione e gerarchia

Un tema ricorrente nella letteratura sui modelli del mondo è la necessità di rappresentazioni gerarchiche. Le rappresentazioni di basso livello catturano dettagli sensoriali fini (colori, texture, frequenze sonore), mentre quelle di livello superiore codificano strutture astratte (oggetti, relazioni, categorie, affordanze). Una pianificazione efficiente richiede di operare principalmente a livelli alti di astrazione, ricorrendo ai livelli inferiori solo quando necessario.

La tensione tra fedeltà e compattezza della rappresentazione si formalizza nel cosiddetto *information bottleneck* di Tishby, Pereira e Bialek (2000). Si consideri una variabile di ingresso X (l'osservazione sensoriale), una variabile target Y (la ricompensa o l'azione ottimale), e una rappresentazione compressa T. Il principio dell'information bottleneck cerca la rappresentazione T che massimizza l'informazione mutua con Y preservandone la quantità minima necessaria da X:

$$\max_{p(T|X)} I(T; Y) - \beta \cdot I(T; X)$$

dove $\beta > 0$ controlla il compromesso tra compressione e pertinenza predittiva. Si può dimostrare che le soluzioni ottimali di questo problema formano una *curva di trade-off informazionale* i cui punti estremi corrispondono rispettivamente alla rappresentazione identità ($\beta \to 0$) e alla statistica sufficiente minimale per Y ($\beta \to \infty$).

L'information bottleneck offre una prospettiva teorica sull'apprendimento delle rappresentazioni: una buona codifica latente di un modello del mondo dovrebbe catturare esattamente le caratteristiche dell'osservazione che sono predittivamente rilevanti per le azioni future, scartando tutto il resto come rumore superfluo. Questa intuizione sarà centrale nell'analisi della JEPA.

II. Formalismo matematico dei modelli a variabile latente: inferenza variazionale, RSSM e Dreamer

1. Autoencoder variazionali: derivazione dell'ELBO

L'autoencoder variazionale (VAE), introdotto da Kingma e Welling nel 2013 e in modo indipendente da Rezende, Mohamed e Wierstra nello stesso anno, costituisce il fondamento matematico della maggior parte dei modelli generativi profondi impiegati nei world model contemporanei. La sua derivazione rigorosa richiede di partire dal problema dell'inferenza bayesiana approssimata.

Si supponga di disporre di un insieme di osservazioni $x_{1:N}$ generate da un processo latente: ogni osservazione x_i è condizionalmente indipendente dalla variabile latente z_i secondo il modello generativo:

$$p_\theta(x, z) = p_\theta(x \mid z) \cdot p(z)$$

dove $p(z)$ è una distribuzione a priori fissata (tipicamente una gaussiana isotropa standard $\mathcal{N}(0, I)$) e $p_\theta(x \mid z)$ è il modello di decodifica parametrizzato da una rete neurale. L'apprendimento dei parametri θ per massima verosimiglianza marginale richiederebbe il calcolo di:

$$\log p_\theta(x) = \log \int p_\theta(x \mid z) \cdot p(z)\, dz$$

Questo integrale è intrattabile in dimensione alta perché la porzione dello spazio z che contribuisce significativamente all'integrale è una regione estremamente piccola e difficile da esplorare con campionamento casuale.

La soluzione variazionale introduce una distribuzione approssimante $q_\varphi(z \mid x)$ — il *codificatore* o *encoder* — e costruisce un limite inferiore alla log-verosimiglianza marginale, noto come Evidence Lower BOund o ELBO:

$$\begin{aligned}
\log p_\theta(x) &= \log \int p_\theta(x \mid z) \cdot p(z)\, dz \\
&= \log \int q_\varphi(z \mid x) \cdot [p_\theta(x \mid z) \cdot p(z) \,/\, q_\varphi(z \mid x)]\, dz \\
&\geq \int q_\varphi(z \mid x) \cdot \log[p_\theta(x \mid z) \cdot p(z) \,/\, q_\varphi(z \mid x)]\, dz \quad \text{(per la disuguaglianza di Jensen)} \\
&= \mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x \mid z)] - KL[q_\varphi(z \mid x) \,||\, p(z)] \\
&=: \mathcal{L}(\theta, \varphi; x)
\end{aligned}$$

Il primo termine è la verosimiglianza di ricostruzione attesa: misura quanto bene il decodificatore riproduce l'osservazione dato un codice latente campionato dall'encoder. Il secondo termine è la divergenza di Kullback-Leibler tra la distribuzione posteriore approssimata e il prior: funge da regolarizzatore che impedisce all'encoder di collassare in una funzione deterministica.

Lo scarto tra il limite inferiore e la log-verosimiglianza vera è esattamente la divergenza KL tra l'approssimazione variazionale e la vera posteriore:

$$\log p_\theta(x) - \mathcal{L}(\theta, \varphi; x) = KL[q_\varphi(z \mid x) \,||\, p_\theta(z \mid x)] \geq 0$$

Massimizzare l'ELBO è quindi equivalente a massimizzare simultaneamente la verosimiglianza marginale e minimizzare la discrepanza tra l'encoder e la vera posteriore. La traduzione in algoritmo differenziabile richiede la gestione del gradiente rispetto a φ del termine di aspettazione. Poiché il campionamento diretto da $q_\varphi(z \mid x)$ non è differenziabile rispetto a φ, si adotta il *reparametrization trick*: se $q_\varphi(z \mid x) = \mathcal{N}(z; \mu_\varphi(x), \mathrm{diag}(\sigma_\varphi^2(x)))$, allora si scrive:

$$z = \mu_\varphi(x) + \sigma_\varphi(x) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I)$$

dove $\odot$ denota il prodotto elemento per elemento (prodotto di Hadamard). In questo modo z dipende deterministicamente dai parametri φ attraverso μ_φ e σ_φ, e il gradiente può fluire attraverso di essi. Il termine KL tra due gaussiane ammette una forma chiusa:

$$KL[\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) \,||\, \mathcal{N}(0, I)] = \tfrac{1}{2} \Sigma_j (\mu_j^2 + \sigma_j^2 - \log \sigma_j^2 - 1)$$

dove la somma scorre sulle dimensioni dello spazio latente. Questa espressione analitica elimina la necessità di stimare il termine di regolarizzazione tramite campionamento Monte Carlo.

2. Il Recurrent State Space Model (RSSM)

L'estensione del framework VAE alle sequenze temporali è il problema centrale dei modelli del mondo profondi. La risposta più influente a questo problema è il *Recurrent State Space Model* (RSSM), introdotto da Hafner e collaboratori nel 2019 all'interno dell'architettura Dreamer e poi raffinato nelle versioni DreamerV2 e DreamerV3.

L'idea fondamentale dell'RSSM è di partizionare lo stato latente in due componenti complementari: uno stato deterministico h_t e uno stato stocastico z_t. Questa separazione serve a catturare sia la continuità deterministica delle traiettorie (affidata a h_t) sia l'incertezza irriducibile sull'evoluzione del mondo (affidata a z_t).

Il modello si articola nelle seguenti distribuzioni:

Modello di transizione deterministico (noto come *sequence model* o *GRU cell*):

$$h_t = f_\varphi(h_{t-1}, z_{t-1}, a_{t-1})$$

dove f_φ è una Gated Recurrent Unit (GRU). Lo stato h_t integra tutta l'informazione deterministica della traiettoria fino all'istante $t - 1$.

Modello di transizione stocastico — distribuzioni a priori e a posteriori:

$$\text{Prior:} \quad z_t \sim p_\varphi(z_t \mid h_t)$$
$$\text{Posterior:} \quad z_t \sim q_\varphi(z_t \mid h_t, o_t)$$

Il prior descrive la distribuzione sullo stato stocastico al tempo t conoscendo solo la storia deterministica h_t; la posteriore incorpora anche l'osservazione corrente o_t.

Modello di osservazione (*decoder*):

$$o_t \sim p_\theta(o_t \mid h_t, z_t)$$

Modello di ricompensa:

$$r_t \sim p_\psi(r_t \mid h_t, z_t)$$

In DreamerV2 e DreamerV3, le distribuzioni stocastiche z_t non sono gaussiane ma categoriali: ogni z_t è un vettore di K variabili categoriali con C categorie ciascuna, per un totale di $K \times C$ possibilità. Questa scelta è motivata dalla constatazione empirica

che le rappresentazioni categoriali risultano più efficienti e stabili per modellare stati del mondo con struttura discreta o multimodale. Il gradiente attraverso variabili categoriali si stima con lo *straight-through estimator* di Bengio o con la tecnica di Jang (Gumbel-Softmax).

3. Derivazione della loss dell'RSSM

La funzione di perdita dell'RSSM si ottiene massimizzando l'ELBO sulla sequenza, che factorizza sulla lunghezza T della traiettoria:

$$\mathcal{L} = \Sigma_{t=1}^{T} \{ \mathbb{E}_{q_\varphi(z_t \mid h_t, o_t)}[\log p_\theta(o_t \mid h_t, z_t) + \log p_\psi(r_t \mid h_t, z_t)] - \beta \cdot KL[q_\varphi(z_t \mid h_t, o_t) \,||\, p_\varphi(z_t \mid h_t)] \}$$

Il parametro $\beta > 0$ controlla il peso del termine di regolarizzazione KL: valori piccoli permettono all'encoder di sfruttare pienamente l'osservazione corrente, ma rischiano di degradare la qualità del prior e quindi la capacità di immaginare (*imagine*) traiettorie coerenti a partire dal solo h_t. DreamerV3 introduce una tecnica di normalizzazione adattiva del segnale che elimina la necessità di selezionare manualmente β.

Un'asimmetria importante nel training dell'RSSM riguarda il flusso del gradiente: durante l'addestramento del modello generativo, il gradiente non fluisce attraverso il campionamento $z_t \sim q_\varphi$, bensì viene gestito tramite riparametrizzazione. Tuttavia, il calcolo di h_t attraverso la GRU richiede z_{t-1}, che durante il rollout immaginario è campionato dal prior piuttosto che dalla posteriore. Questo crea una discrepanza tra distribuzione di addestramento e distribuzione di inferenza (*train-test mismatch*), parzialmente mitigata nella variante scheduled sampling.

4. Pianificazione e comportamento nell'ambiente immaginario

Una volta addestrato il modello del mondo, Dreamer addestra un attore e un critico interamente nell'ambiente immaginario, senza interagire con il mondo reale durante questa fase. Il rollout immaginario di lunghezza H si svolge come segue:

Inizializzazione: (h_t, z_t) dallo stato posteriore sull'osservazione reale più recente

Per $\tau = t, ..., t + H - 1$:

$a_\tau \sim \pi_\omega(a_\tau \mid h_\tau, z_\tau)$ — attore (politica)

$h_{\tau+1} = f_\varphi(h_\tau, z_\tau, a_\tau)$ — GRU deterministico

$z_{\tau+1} \sim p_\varphi(z_{\tau+1} \mid h_{\tau+1})$ — prior stocastico

$r_{\tau+1} \sim p_\psi(r_{\tau+1} \mid h_{\tau+1}, z_{\tau+1})$

Il valore dell'orizzonte τ è stimato combinando il ritorno scontato dei reward immaginari con una stima di valore appresa da un critico V_ξ, nella forma della stima λ-ritorno:

$$V_\tau^\lambda = r_{\tau+1} + \gamma [(1 - \lambda) V_\xi(h_{\tau+1}, z_{\tau+1}) + \lambda V_{\tau+1}^\lambda]$$

dove γ è il fattore di sconto e $\lambda \in [0,1]$ controlla il bias-varianza trade-off dell'estimatore ($\lambda = 0$ produce il TD target a un passo, $\lambda = 1$ il Monte Carlo return).

L'attore viene addestrato massimizzando i valori attesi lungo il rollout immaginario. Poiché il rollout è differenziabile rispetto ai parametri ω (attraverso il reparametrization trick per le variabili stocastiche z_τ e la Gumbel-Softmax per le categoriali), si può retropropagare direttamente attraverso i passi temporali immaginari — una tecnica denominata *analytic gradients through imagination*. Il critico, invece, viene addestrato minimizzando l'errore quadratico medio tra $V_\xi(h_\tau, z_\tau)$ e i target V_τ^λ.

5. Analisi della convergenza e stabilità numerica

Un aspetto spesso trascurato nella letteratura applicata riguarda le condizioni di stabilità numerica del training dell'RSSM. La retropropagazione attraverso T passi della GRU soffre del problema del gradiente che svanisce o esplode (*vanishing/exploding gradient*), analogamente ad ogni rete ricorrente profonda. Hafner e collaboratori in DreamerV3 adottano due strategie chiave.

La prima è la normalizzazione simmetrica dello stato KL, che rimpiazza la divergenza KL asimmetrica nella loss con una combinazione:

$$KL_{sym}(z_t) = \alpha \cdot KL[sg(q_\varphi(z_t \mid h_t, o_t)) \,||\, p_\varphi(z_t \mid h_t)] + (1 - \alpha) \cdot KL[q_\varphi(z_t \mid h_t, o_t) \,||\, sg(p_\varphi(z_t \mid h_t))]$$

dove $sg(\cdot)$ denota l'operazione *stop-gradient* (il cui gradiente viene bloccato), e $\alpha = 0.8$ nel paper. Questo schema aggiorna il prior e il posterior in proporzioni diverse, migliorando la stabilità dell'addestramento.

La seconda strategia è la normalizzazione dei reward attraverso una stima esponenzialmente decadente della loro scala:

$$r_t^{norm} = (r_t - \mu_r) / \max(1, S_r)$$

dove μ_r è la media mobile dei reward e S_r è la deviazione standard mobile. Questa normalizzazione rende Dreamer applicabile a task con scale di reward molto diverse senza necessità di tuning degli iperparametri — proprietà denominata dagli autori *robustness across domains*.

6. Modelli del mondo a base di Transformer: TWM e iris

La sostituzione della GRU con un'architettura Transformer nel modello di transizione stocastico ha stimolato una nuova generazione di world model. Il *Transformer World Model* (TWM) di Hafner e collaboratori, e l'architettura IRIS di Micheli e colleghi (2022), mostrano che l'attenzione multi-testa (*multi-head self-attention*) modella più efficacemente le dipendenze a lungo termine nelle sequenze.

IRIS, in particolare, tratta la sequenza di token latenti (ottenuti da un VQVAE — Vector Quantized VAE) come una sequenza di linguaggio e la modella con un decoder Transformer autoregressivo. Il meccanismo di attenzione scala quadraticamente con la lunghezza del contesto, ma all'interno dei world model le traiettorie hanno lunghezze moderate (decine di passi), rendendo questo costo computazionale accettabile.

La funzione di attenzione scalata dell'architettura Transformer è:

$$\text{Attn}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d_k}) \cdot V$$

dove $Q \in \mathbb{R}^{n \times d_k}$, $K \in \mathbb{R}^{m \times d_k}$, $V \in \mathbb{R}^{m \times d_v}$ sono rispettivamente le matrici di query, chiave e valore, n è la lunghezza della sequenza di query, m quella della sequenza di chiave-valore, e d_k è la dimensione delle chiavi. Il fattore di scala $1/\sqrt{d_k}$ previene la saturazione della softmax in alta dimensione.

In IRIS, l'autoencoder VQVAE quantizza ogni frame o_t in una sequenza di K_{patch} token discreti, ciascuno tratto da un codebook $\mathscr{C} = \{e_1, ..., e_N\}$ con $e_i \in \mathbb{R}^d$. La quantizzazione opera attraverso il *nearest-neighbor lookup*:

$$z_q = \text{argmin}_{e \in \mathscr{C}} ||z_e - e||_2$$

dove $z_e = \text{Enc}(o_t)$ è la codifica continua del frame. Il gradiente viene trasportato attraverso la discontinuità del argmin con il straight-through estimator. La loss del VQVAE è:

$$\mathcal{L}_{VQVAE} = ||o_t - \text{Dec}(z_q)||_2^2 + ||\text{sg}(z_e) - z_q||_2^2 + \beta_{commit} \cdot ||z_e - \text{sg}(z_q)||_2^2$$

Il primo termine è la loss di ricostruzione, il secondo spinge il codebook verso le codifiche, il terzo (*commitment loss*) spinge le codifiche verso il codebook selezionato.

7. Codifiche strutturate e modelli a grafo

Un'ulteriore direzione di ricerca, cui è opportuno dedicare una riflessione formale, riguarda l'introduzione di struttura esplicita nelle rappresentazioni latenti. Invece di

modellare lo stato del mondo come un vettore monolitico, si propone di suddividerlo in un insieme di entità con proprietà e relazioni. Questa visione è incarnata dai *Graph Neural Network World Model* (GNWM) e, in forma più astratta, dai modelli a slot (*slot-based models*) come Slot Attention.

In un modello a grafo, lo stato latente è un grafo G_t = (V_t, E_t) dove V_t = {v_t^1, ..., v_t^n} è l'insieme dei nodi (entità) e E_t è l'insieme degli archi (relazioni). La transizione avviene attraverso un *message-passing* step:

m_t^{ij} = φ_e(v_t^i, v_t^j, e_t^{ij}) — messaggio dall'entità j all'entità i
v_{t+1}^i = φ_v(v_t^i, Σ_{j ∈ $\mathcal{N}$(i)} m_t^{ij}) — aggiornamento del nodo

dove φ_e e φ_v sono funzioni neurali condivise tra tutti gli archi e tutti i nodi rispettivamente (weight tying), e $\mathcal{N}$(i) è il vicinato del nodo i nel grafo. Questa struttura garantisce equivarianza rispetto alla permutazione delle entità, proprietà fondamentale per la generalizzazione a nuove configurazioni di oggetti.

I modelli strutturati presentano vantaggi potenziali di generalizzazione e interpretabilità rispetto ai modelli monolitici, ma la loro applicazione pratica ai task di visione è complicata dalla necessità di estrarre automaticamente le entità dai frame — un problema di *object discovery* che rimane aperto. Le architetture Slot Attention (Locatello et al., 2020) e le sue varianti offrono meccanismi di binding attenzione-oggetto differenziabili, ma la qualità della decomposizione degrada su scene reali ad alta complessità visiva.

III. JEPA: architettura, fondamenti energetici e varianti visive

1. Il manifesto di LeCun e la critica ai modelli generativi

Nel gennaio 2022 Yann LeCun rese pubblico un documento tecnico dal titolo "A Path Towards Autonomous Machine Intelligence", destinato a diventare uno dei testi programmatici più discussi nell'intera comunità dell'apprendimento automatico. Il documento non si limita a proporre una nuova architettura, bensì articola una critica sistematica alle paradigmi dominanti — apprendimento per rinforzo, modelli linguistici autoregressivi, modelli generativi per pixel — e ne individua i limiti strutturali rispetto all'intelligenza autonoma di tipo animale.

La critica più acuta riguarda i modelli generativi che operano nello spazio degli input. Un modello che impara a prevedere ogni pixel di un fotogramma futuro deve devolvere enorme capacità rappresentativa alla codifica di dettagli irrilevanti per il controllo: la texture precisa delle foglie di un albero sullo sfondo, le variazioni di

illuminazione su superfici uniformi, il rumore di sensore. Questa esuberanza descrittiva non è solo computazionalmente costosa; essa introduce, secondo LeCun, un problema fondamentale di *mode coverage*: un modello generativo probabilistico deve assegnare massa di probabilità a ogni possibile osservazione futura coerente con la storia passata e l'azione intrapresa. Quando il futuro è genuinamente incerto — come accade quasi sempre nel mondo fisico — il modello è costretto o a collassare su una singola previsione (con conseguente sfocatura per regressione verso la media) o a parametrizzare l'intera distribuzione multi-modale sugli input ad alta dimensione, problema NP-arduo in generale.

La proposta alternativa di LeCun si radica nella tradizione dei modelli a base energetica (*energy-based models*, EBM) e nella architettura denominata JEPA (*Joint Embedding Predictive Architecture*). L'idea di fondo è che un sistema intelligente non debba prevedere la rappresentazione sensoriale grezza del futuro, bensì la sua rappresentazione astratta in uno spazio latente opportunamente appreso. In questo spazio latente, la previsione opera su quantità semanticamente significative e scarta, per costruzione, i gradi di libertà irrilevanti. La domanda che sorge immediatamente — come impedire al sistema di collassare su una rappresentazione costante o degenere — è il cuore tecnico dell'intera proposta.

2. Modelli a base energetica: definizione e proprietà

Un *energy-based model* definisce una funzione di compatibilità scalare $F: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ tale che $F(x, y)$ sia bassa quando x e y sono compatibili secondo il modello e alta altrimenti. La distribuzione di probabilità implicita associata a F è la distribuzione di Gibbs-Boltzmann:

$$p_F(y \mid x) = \exp(-F(x, y)) \,/\, Z(x)$$

dove $Z(x) = \int \exp(-F(x, y))\, dy$ è la funzione di partizione, in genere intrattabile. Il vantaggio degli EBM rispetto ai modelli probabilistici espliciti è che non richiedono di calcolare né di normalizzare $Z(x)$: la previsione consiste semplicemente nel trovare la y che minimizza l'energia:

$$\hat{y} = \operatorname{argmin}_{y \in \mathcal{Y}} F(x, y)$$

L'addestramento di un EBM richiede che la funzione di energia sia bassa sui dati reali e alta sulle configurazioni non osservate (*negative samples*). Formalmente, la loss contrastiva su un esempio positivo (x, y) e un insieme di esempi negativi $\{y^-_k\}$ può essere scritta come:

$$\mathcal{L}_{EBM} = F(x, y) + \log \Sigma_k \exp(-F(x, y^-_k))$$

Questa espressione è la *contrastive loss* o *InfoNCE* quando i negativi sono campionati uniformemente. Se la funzione di partizione fosse trattabile, la minimizzazione di $\mathcal{L}$_EBM rispetto ai parametri di F convergerebbe alla stima di massima verosimiglianza. In pratica, la scelta e la generazione dei campioni negativi costituisce la sfida principale.

Gli EBM possono essere *latenti* quando si introduce una variabile latente z che parametrizza le diverse modalità della distribuzione:

$$F(x, y) = \min_{z \in \mathcal{Z}} E(x, y, z)$$

In questo caso la funzione di energia viene minimizzata sulle variabili latenti prima di confrontare x e y. Il modello JEPA è un EBM latente definito nello spazio delle rappresentazioni, non degli input.

3. Architettura JEPA: formalizzazione rigorosa

Si consideri una coppia (x, y) dove x è il contesto (ad esempio una porzione di immagine o una finestra temporale) e y è il segnale target (ad esempio la porzione complementare dell'immagine o il fotogramma futuro). La JEPA definisce:

Encoder del contesto: $s_x = f_\theta(x)$ $\quad s_x \in \mathbb{R}^{d_s}$
Encoder del target: $s_y = g_\psi(y)$ $\quad s_y \in \mathbb{R}^{d_s}$
Predittore: $\hat{s}_y = h_\varphi(s_x, z)$ $\quad \hat{s}_y \in \mathbb{R}^{d_s}$

dove $z \in \mathbb{R}^{d_z}$ è una variabile latente opzionale che codifica l'incertezza irriducibile sulla y nel caso in cui il contesto x non la determini univocamente. La funzione di energia della JEPA è:

$$F_{JEPA}(x, y, z) = D(\hat{s}_y, s_y) = D(h_\varphi(f_\theta(x), z), g_\psi(y))$$

dove D è una misura di dissimilarità, tipicamente la distanza euclidea al quadrato o la distanza coseno. La JEPA addestrata con questa loss tende — senza accorgimenti ulteriori — verso il collasso rappresentazionale: la soluzione banale in cui f_θ, g_ψ e h_φ producono costantemente lo stesso vettore ha energia $F = 0$ su ogni coppia (x, y). Il problema del collasso è dunque il problema centrale di progettazione di ogni JEPA.

4. Strategie anti-collasso: stop-gradient, momentum encoder e regolarizzazione della varianza

Le strategie proposte per prevenire il collasso delle rappresentazioni si suddividono in tre famiglie concettuali distinte, ciascuna con caratterizzazione matematica propria.

Stop-gradient con predittore asimmetrico (SimSiam, Chen & He 2021): si propaga il gradiente solo attraverso uno dei due rami dell'architettura. Formalmente, la loss è:

$$\mathcal{L}_{SimSiam} = - \hat{s}_y \cdot sg(s_y) \,/\, (||\hat{s}_y||_2 \cdot ||sg(s_y)||_2)$$

dove $sg(\cdot)$ blocca il flusso del gradiente. In questo schema, f_θ non riceve gradiente diretto dalla distanza tra le rappresentazioni, e il predittore h_φ è costretto a imparare una trasformazione non banale per approssimare le rappresentazioni del ramo target. L'analisi teorica di Chen e He mostra che, alla convergenza, lo stop-gradient induce implicitamente una forma di aspettazione sugli aumenti di dati, prevenendo il collasso senza necessità di esempi negativi espliciti.

Momentum encoder (MoCo, BYOL): l'encoder del target g_ψ non viene aggiornato con la discesa del gradiente, bensì come media esponenzialmente pesata dei parametri dell'encoder del contesto f_θ:

$$\psi \leftarrow m \cdot \psi + (1 - m) \cdot \theta, \quad m \in [0.99, 0.999]$$

Questo schema introduce un ritardo temporale nella propagazione delle modifiche parametriche al ramo target, creando un bersaglio in movimento che — in modo non ovvio ma dimostrabile — non collassa. L'analisi di Tian, Chen e Ganguli (2021) mostra che il momentum encoder implementa implicitamente una forma di *Expectation-Maximization*: l'encoder del contesto svolge il passo E (inferenza della rappresentazione ottimale dato il target corrente), mentre l'aggiornamento del momentum svolge il passo M (aggiornamento dei parametri del modello data la rappresentazione corrente).

Regolarizzazione della varianza e covarianza (VICReg, Bardes, Ponce, LeCun 2022): invece di operare tramite asimmetria architetturale, VICReg aggiunge esplicitamente termini di regolarizzazione che impongono al codificatore di produrre rappresentazioni con varianza elevata e covarianza ridotta. La loss si compone di tre termini:

$$\mathcal{L}_{VICReg} = \lambda \cdot \ell_{inv} + \mu \cdot \mathcal{L}_{var} + \nu \cdot \ell_{cov}$$

dove:

$$\ell_{inv} = (1/N) \sum_{i=1}^{N} ||z_i - z'_i||_2^2 \quad \text{— invarianza}$$
$$\mathcal{L}_{var} = (1/d) \sum_{j=1}^{d} \max(0, \gamma - S(z_j, \varepsilon)) \quad \text{— varianza}$$
$$\ell_{cov} = (1/(d(d-1))) \sum_{i \neq j} [Cov(Z)_{ij}]^2 \quad \text{— covarianza}$$

con $S(z_j, \varepsilon) = \sqrt{(Var(z_j) + \varepsilon)}$ la deviazione standard del j-esimo canale, $\gamma = 1$ un obiettivo minimo di varianza, e $Cov(Z) = (1/(N-1)) Z^T Z$ la matrice di covarianza empirica calcolata su un batch di N campioni (con Z centrata). Il termine di varianza

impedisce che i canali collassino a valori costanti; il termine di covarianza spinge le dimensioni dello spazio latente ad essere decorrelate, promuovendo l'uso efficiente della capacità rappresentativa.

VICReg è particolarmente interessante da un punto di vista teorico perché massimizzare ℓ_var e minimizzare ℓ_cov equivale, nel limite continuo, a promuovere una distribuzione delle rappresentazioni con matrice di covarianza proporzionale all'identità — una forma di *whitening* implicito. Questa condizione è strettamente collegata alla massimizzazione dell'entropia delle rappresentazioni soggetta a vincolo di norma, che a sua volta corrisponde alla regolarizzazione dell'information bottleneck discussa nella Parte I.

5. I-JEPA: predizione nello spazio latente visivo

I-JEPA (*Image Joint Embedding Predictive Architecture*), presentata da Assran e collaboratori (Meta AI, 2023), è la prima implementazione su larga scala del paradigma JEPA applicato alla comprensione delle immagini statiche. L'architettura segue la struttura generale descritta nella sezione 3, con le seguenti specificità.

Il contesto x è ottenuto mascherando blocchi dell'immagine originale: vengono selezionati casualmente K = 4 rettangoli target e una regione di contesto complementare, ottenuta rimuovendo dall'immagine intera tutti i token corrispondenti ai target. Sia ViT (*Vision Transformer*) l'encoder condiviso: esso processa la patch-sequence dell'immagine attraverso blocchi di attenzione multi-testa, producendo per ogni patch (token) un vettore di rappresentazione.

L'encoder del contesto f_θ è un ViT standard che riceve i token del contesto (posizione degli embedding posizionali preservata) e produce le loro rappresentazioni. L'encoder del target g_ψ è anch'esso un ViT, aggiornato con momentum rispetto a f_θ (schema EMA). Per ogni regione target, il predittore h_φ riceve la rappresentazione del contesto e i token posizionali della regione target (senza i valori reali dei patch), e deve produrre le rappresentazioni target nel tuo spazio latente:

$$\hat{s}_{y_k} = h_\varphi(s_x, pos_{y_k}), \quad k = 1, ..., K$$

La loss è la distanza euclidea media tra previsioni e target nel spazio latente normalizzato:

$$\mathcal{L}_{IJEPA} = (1/K) \Sigma_{k=1}^{K} (1/|y_k|) \Sigma_{p \in y_k} ||\hat{s}_y^{(p)} / ||\hat{s}_y^{(p)}||_2 - s_y^{(p)} / ||s_y^{(p)}||_2||_2^2$$

dove |y_k| è il numero di patch nella k-esima regione target. La normalizzazione L2 è fondamentale per stabilizzare l'addestramento quando si usa EMA senza stop-gradient simmetrico.

Il risultato sorprendente di I-JEPA è che, addestrando con questo obiettivo puramente di predizione latente (senza alcuna supervisione esplicita, senza augmentation aggressive, senza decodificatore che ricostruisce i pixel), il ViT-H/14 addestrato con I-JEPA su ImageNet raggiunge prestazioni comparabili o superiori a DINO e MAE su downstream tasks lineari, pur richiedendo un ordine di grandezza meno di augmentation. La qualità delle rappresentazioni ottenute suggerisce che la predizione latente impone constraint molto più ricchi sulla struttura delle feature rispetto alla semplice ricostruzione dei pixel mascherati.

6. V-JEPA: estensione alla dimensione temporale

V-JEPA (*Video Joint Embedding Predictive Architecture*), proposta da Bardes e collaboratori (Meta AI, 2024), estende il framework al dominio video con una modifica concettualmente semplice ma tecnicamente impegnativa: il masking viene applicato nello spazio-tempo 3D e la previsione avviene su blocchi spazio-temporali.

Dato un video $V \in \mathbb{R}^{T \times H \times W \times C}$, esso viene suddiviso in un grigliato di tube spazio-temporali. Il masking seleziona casualmente un sottoinsieme di tube da predire; il contesto è il complemento. In V-JEPA, lo stesso ViT 3D processa sia il contesto sia il target, con embedding posizionali spazio-temporali fattorisizati:

$$\text{pos}(t, h, w) = \text{pos}_t(t) + \text{pos}_h(h) + \text{pos}_w(w)$$

La scelta di masking è cruciale: V-JEPA impiega tube continue (contiguous tube masking) invece di patch isolate, obbligando il modello a predire interi blocchi di movimento coerente. La fraction di masking è molto elevata — tipicamente 90% — il che rende il problema genuinamente sfidante.

La funzione di loss di V-JEPA mantiene la struttura di I-JEPA:

$$\mathcal{L}_{\text{VJEPA}} = (1/|\mathcal{M}|) \Sigma_{p \in \mathcal{M}} ||\hat{s}_V^{(p)} / ||\hat{s}_V^{(p)}||_2 - s_V^{(p)} / ||s_V^{(p)}||_2||_2^2$$

dove $\mathcal{M}$ è l'insieme dei token mascherati e $s_V^{(p)}$, $\hat{s}_V^{(p)}$ sono rispettivamente le rappresentazioni target (dall'encoder EMA) e le previsioni del predittore. La chiave teorica è che il predittore deve apprendere un modello implicito della dinamica spazio-temporale: esso non ha accesso ai valori reali dei token da predire, solo ai token contestuali e alle posizioni dei target, dunque deve costruire internamente una stima delle traiettorie di movimento, delle relazioni causali e delle invarianze temporali.

Rispetto ai generative video model (come i video diffusion model), V-JEPA non richiede di produrre pixel realistici, ma solo rappresentazioni coerenti nel proprio spazio latente. Questo lo rende computazionalmente molto più parco e concettualmente allineato con la tesi di LeCun: la comprensione del mondo non

richiede la capacità di immaginare ogni dettaglio sensoriale, bensì di anticipare le trasformazioni semanticamente rilevanti.

7. Il problema della completezza semantica e il legame con la casualità

Un punto delicato, spesso trascurato, riguarda la *completezza semantica* delle rappresentazioni apprese da una JEPA. Poiché la JEPA non è vincolata a ricostruire l'input, non vi è garanzia a priori che le sue rappresentazioni latenti catturino tutte le informazioni presenti nell'osservazione originale. Potrebbero, in linea di principio, scartare anche informazioni che risultano rilevanti per compiti a valle (*downstream tasks*) ma che non sono predittivamente utili nel task di pretraining.

Formalmente, la rappresentazione s = f_θ(x) apprende la statistica sufficiente minimale per predire s_y = g_ψ(y) dal contesto x, il che equivale a massimizzare I(s_x; s_y) soggetto a compressione. Non vi è motivo teorico per cui questa statistica coincida con la statistica sufficiente minimale per ogni task a valle y'. Se la struttura del masking non è sufficientemente varia da coprire le dipendenze rilevanti per i task a valle, le rappresentazioni potrebbero risultare sub-ottimali in transfer learning.

In pratica, I-JEPA e V-JEPA mostrano che la predizione latente di patch/tube ampi e contigui produce rappresentazioni generaliste di alta qualità, suggerendo che la struttura spaziale e temporale del masking funge da inductive bias statisticamente ricco. Tuttavia, la questione teorica rimane aperta e costituisce un fronte attivo di ricerca: come caratterizzare formalmente la relazione tra la struttura del task di pretraining e la qualità delle rappresentazioni per compiti a valle arbitrari?

8. Gerarchia di JEPA e il modello architettonico completo di LeCun

Il progetto di LeCun non si esaurisce in una singola JEPA: l'architettura autonoma completa proposta nel documento del 2022 prevede una gerarchia di JEPA annidate, ciascuna operante a un diverso livello di astrazione spazio-temporale. Al livello più basso vi sono le JEPA che modellano le transizioni sensoriali immediate (frame-to-frame); ai livelli intermedi vi sono moduli che predicono stati astratti su scale temporali più lunghe (sequenze di azioni); al livello più alto vi è un modulo di pianificazione che opera su rappresentazioni simboliche o quasi-simboliche.

La pianificazione in questa gerarchia è concepita come ottimizzazione dell'energia lungo una sequenza di stati latenti. Dato uno stato corrente s_0 e uno stato obiettivo s_G, il pianificatore cerca la sequenza di stati intermedi s_1, ..., s_{H} e di azioni a_0, ..., a_{H-1} che minimizzano la funzione di costo totale:

$$\mathscr{C}(s_{0:H}, a_{0:H-1}) = \Sigma_{t=0}^{H-1} c(s_t, a_t) + c_G(s_H, s_G)$$

soggetto ai vincoli imposti dai modelli di transizione di livello superiore:

$s_{t+1} \approx \text{JEPA_level}(s_t, a_t)$

Questo schema è essenzialmente un MPC (*Model Predictive Control*) gerarchico nello spazio latente, e la sua realizzazione pratica richiede che i modelli di transizione siano sufficientemente accurati e differenziabili da permettere l'ottimizzazione della sequenza di azioni tramite gradient descent. La JEPA, essendo differenziabile end-to-end, si presta naturalmente a questo schema.

IV. Robotica incarnata, controllo model-based e pianificazione nello spazio latente

1. L'ipotesi dell'incarnazione e i suoi fondamenti teorici

La robotica contemporanea, intesa come campo che studia il comportamento di agenti fisici nel mondo reale, ha intrattenuto con la teoria dei modelli del mondo un rapporto storicamente tormentato. Da un lato, la tradizione del controllo classico si fonda su modelli espliciti e accurati della dinamica del robot e dell'ambiente; dall'altro, l'apprendimento automatico ha a lungo promesso di superare questa dipendenza attraverso politiche apprese direttamente dall'esperienza sensorimotoria. La sintesi emergente — il controllo model-based fondato su modelli appresi — è oggi il paradigma più ricco e teoricamente fondato.

L'ipotesi dell'incarnazione (*embodiment hypothesis*), formulata in ambito cognitivo da Varela, Thompson e Rosch nel 1991 e ripresa in chiave computazionale da Brooks, Pfeifer e Bongard, sostiene che l'intelligenza emerge non dalla manipolazione astratta di simboli, bensì dall'interazione continua e ricorsiva tra un corpo fisicamente situato e il suo ambiente. Questa prospettiva ha implicazioni profonde per la progettazione di sistemi di apprendimento: il modello del mondo non è una rappresentazione neutra dell'ambiente esterno, bensì una struttura relazionale che codifica le *affordanze* — le possibilità di azione — rilevanti per quel corpo specifico in quell'ambiente specifico.

La nozione di affordanza, introdotta dallo psicologo ecologico James Gibson nel 1979, è formalizzabile in termini di modelli del mondo come la funzione che mappa lo stato del mondo s e le proprietà morfologiche del corpo b in un insieme di azioni eseguibili $\mathcal{A}(s, b) \subseteq \mathcal{A}$. Un robot con attuatori diversi da quelli di un umano avrà affordanze radicalmente differenti anche a fronte dello stesso stato del mondo. Questo implica che i modelli del mondo appresi sono necessariamente *agent-centric*: la loro validità è relativa alla morfologia e alle capacità motorie dell'agente che li ha costruiti.

2. Dinamica del robot: modellazione differenziabile

La dinamica di un robot con n gradi di libertà è descritta dall'equazione di Lagrange-Eulero:

$$M(q)\,\ddot{q} + C(q,\dot{q})\,\dot{q} + g(q) = \tau + J^{T}(q)\,F_{ext}$$

dove $q \in \mathbb{R}^n$ è il vettore delle coordinate generalizzate (angoli dei giunti), $\dot{q}$ e $\ddot{q}$ le loro derivate prime e seconde, $M(q) \in \mathbb{R}^{n\times n}$ è la matrice di inerzia simmetrica definita positiva, $C(q,\dot{q}) \in \mathbb{R}^{n\times n}$ è la matrice delle forze di Coriolis e centrifughe, $g(q) \in \mathbb{R}^n$ è il vettore delle forze gravitazionali, $\tau \in \mathbb{R}^n$ è il vettore delle coppie attuate, $J(q) \in \mathbb{R}^{6\times n}$ è la matrice jacobiana dello spazio operativo, e $F_{ext} \in \mathbb{R}^6$ il vettore delle forze e momenti esterni. La matrice di inerzia soddisfa:

$$M(q) = \sum_{i=1}^{n} \left[m_i\, J_{v_i}^{T}(q)\, J_{v_i}(q) + J_{\omega_i}^{T}(q)\, R_i\, I_i\, R_i^{T}\, J_{\omega_i}(q) \right]$$

dove m_i è la massa del link i, J_{v_i} e J_{ω_i} sono le colonne del jacobiano traslazionale e rotazionale relative al link i, R_i è la matrice di rotazione del frame del link, e I_i è il tensore di inerzia del link nel suo frame di corpo.

La difficoltà di apprendere questa dinamica da dati risiede nel fatto che le proprietà fisiche del robot — masse, momenti di inerzia, attrito nei giunti, elasticità dei cavi e delle trasmissioni — non sono mai perfettamente note, e si modificano nel tempo a causa di usura, carico variabile o danneggiamenti. Il paradigma dell'*identificazione sistematica* della dinamica tramite esperimenti di eccitazione ottimale costituisce l'approccio classico; quello delle reti neurali differenziabili fisicamente informate (*physics-informed neural networks*, PINN) ne rappresenta l'estensione moderna.

Le PINN per la dinamica robotica parametrizzano le funzioni M, C, g con reti neurali che rispettano le proprietà fisiche strutturali: M deve essere simmetrica definita positiva, C deve soddisfare la proprietà di antisimmetria $\dot{M} - 2C$ = antisimmetrica, e g deve derivare da un potenziale scalare. Questi vincoli vengono imposti sia attraverso l'architettura della rete (ad esempio, parametrizzando M come $L\,L^{T} + \varepsilon I$ dove L è triangolare inferiore) sia attraverso penalità nella loss.

3. Model Predictive Control nello spazio latente

Il *Model Predictive Control* (MPC) è la strategia di controllo ottimo a orizzonte mobile che, ad ogni istante di controllo t, risolve un problema di ottimizzazione a orizzonte finito H e applica solo la prima azione ottimale, ripetendo il processo al passo successivo. La formulazione standard del MPC è:

$$\min_{a_{t:t+H-1}} \Sigma_{\tau=t}^{t+H-1} c(s_\tau, a_\tau) + V_f(s_{t+H})$$

soggetto a:

$$s_{\tau+1} = f(s_\tau, a_\tau), \quad \tau = t, \ldots, t+H-1$$
$$s_\tau \in \mathcal{S}, \quad a_\tau \in \mathcal{A}, \quad \tau = t, \ldots, t+H-1$$

dove $c: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ è il costo corrente, $V_f: \mathcal{S} \to \mathbb{R}$ è la funzione di costo terminale, $f: \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ è il modello di transizione, e $\mathcal{S}, \mathcal{A}$ sono gli insiemi ammissibili di stati e azioni. Le proprietà di stabilità e convergenza del MPC sono ben studiate nella letteratura del controllo: in particolare, l'aggiunta di un costo terminale appropriato e di un insieme terminale di invarianza garantisce la stabilità asintotica della traiettoria chiusa sotto ipotesi di modello esatto.

Quando il modello f è una rete neurale differenziabile appresa dai dati, il problema di ottimizzazione (detto *trajectory optimization* o *shooting method*) si risolve propagando i gradienti del costo rispetto alle azioni attraverso i passi del modello (*backpropagation through time*, BPTT):

$$\partial \mathcal{L} / \partial a_t = \Sigma_{\tau=t}^{H-1} (\partial \mathcal{L}_\tau / \partial s_{\tau+1}) \cdot (\partial s_{\tau+1} / \partial a_t)$$

dove il termine $\partial s_{\tau+1} / \partial a_t$ si fattorizza ricorsivamente come:

$$\partial s_{\tau+1} / \partial a_t = (\partial s_{\tau+1} / \partial s_\tau) \cdot (\partial s_\tau / \partial a_t) \quad \text{per } \tau > t$$
$$\partial s_{t+1} / \partial a_t = \partial f / \partial a \,|_{(s_t, a_t)} \quad \text{per } \tau = t$$

Il prodotto iterato di jacobiani $\partial s_{\tau+1} / \partial s_\tau$ introduce il già menzionato problema del gradiente che si svanisce o esplode per orizzonti lunghi. L'algoritmo MPPI (*Model Predictive Path Integral*) di Williams e collaboratori evita questo problema campionando stocasticamente sequenze di azioni e stimando il gradiente con una media pesata delle traiettorie campionate:

$$a_t^* \approx \Sigma_k w_k \cdot a_t^{(k)} \,/\, \Sigma_k w_k, \quad w_k = \exp(-\mathcal{L}(\tau_k) / \lambda)$$

dove τ_k è la k-esima traiettoria campionata, $\mathcal{L}(\tau_k)$ il suo costo, e $\lambda > 0$ un parametro di temperatura. MPPI è eseguibile su GPU in modo altamente parallelizzato e non richiede la differenziabilità del modello, ma la sua efficienza dipende fortemente dal numero di traiettorie campionate e dalla temperatura λ.

Un risultato empirico recente che merita di essere citato esplicitamente in questa sede riguarda il vantaggio computazionale delle rappresentazioni object-centric rispetto a quelle patch-based nel contesto della pianificazione MPC. C-JEPA, confrontato con DINO-WM — un world model che impiega le rappresentazioni di DINOv2 a livello di patch — sul task di manipolazione Push-T, ottiene performance comparabili utilizzando soltanto l'1% dei token rispetto a DINO-WM e riducendo il tempo di

pianificazione di oltre 8 volte. Questo dato quantifica con precisione il vantaggio computazionale che ci si attende dalla teoria: le rappresentazioni object-centric, comprimendo la scena in un numero di entità proporzionale al numero di oggetti piuttosto che al numero di patch dell'immagine, riducono drasticamente la dimensionalità dello spazio su cui il pianificatore MPC deve operare, rendendo la pianificazione in tempo reale accessibile su hardware embedded. L'argomento è direttamente rilevante per i sistemi robotici incarnati che devono operare sotto vincoli di latenza stretti, e rafforza la tesi che la struttura delle rappresentazioni — non solo la loro accuratezza predittiva — determina la qualità del controllo incarnato.

4. Dreamer applicato alla robotica: PlaNet e successori

L'architettura PlaNet di Hafner e collaboratori (2019) è stata la prima dimostrazione convincente che un robot (simulato) può imparare a controllare un sistema fisico complesso — manipolazione, locomozione, navigazione — usando esclusivamente il modello del mondo come surrogate per la pianificazione, senza mai addestrate una politica parametrica esplicita. PlaNet combina l'RSSM (sezione II.2) con un pianificatore a campionamento CEM (*Cross-Entropy Method*) nello spazio delle azioni.

Il CEM è un algoritmo di ottimizzazione stocastica che mantiene una distribuzione guida su sequenze di azioni — inizialmente gaussiana isotropa — e la aggiorna iterativamente selezionando le sequenze a basso costo:

Inizializzazione: p_0(a_{t:t+H}) = $\mathcal{N}$(0, I)

Per j = 1, ..., J_iter:
  Campionare M sequenze {a^{(k)}_{t:t+H}}_{k=1}^M da p_{j-1}
  Rollare il modello RSSM per ciascuna sequenza e calcolare i costi {$\mathcal{L}$_k}
  Selezionare le N_elite < M sequenze a costo inferiore
  Aggiornare p_j come gaussiana fittata alle N_elite sequenze elette

La semplicità di CEM e la sua indipendenza dal gradiente del modello lo rendono robusto a discontinuità e alla stochasticitá della transizione. Tuttavia, la sua efficienza degrada esponenzialmente con la dimensione dello spazio delle azioni e la lunghezza dell'orizzonte, limitando PlaNet a sistemi con azione a bassa dimensione (tipicamente <6 gradi di libertà).

Le architetture Dreamer sostituiscono il pianificatore CEM con una politica parametrica addestrata nell'ambiente immaginario, come descritto nella Parte II. L'applicazione alla robotica reale introduce ulteriori complicazioni rispetto ai task simulati: il ritardo attuativo (*actuator delay*), il rumore di misura dei sensori reali, la deformazione del modello per stati al di fuori della distribuzione di addestramento, e la fragilità delle politiche rispetto a perturbazioni fisiche non presenti in simulazione.

5. Il problema del sim-to-real transfer e la randomizzazione del dominio

Il *sim-to-real gap* è la discrepanza tra le prestazioni di una politica in simulazione e quelle osservate sul robot fisico. Essa origina da tre sorgenti principali: errori di modellazione della fisica (attrito, deformabilità dei materiali, dinamica dei contatti), errori di modellazione della percezione (rumore del sensore, aberrazioni ottiche, variazioni di illuminazione), e differenze morfologiche tra il modello URDF e il robot fisico (tolleranze di fabbricazione, usura).

La *domain randomization* (Tobin et al., 2017; Andrychowicz et al., 2020 per l'applicazione Dexterous Manipulation) è la strategia più consolidata per mitigare il sim-to-real gap: durante il training in simulazione, i parametri del simulatore vengono perturbati casualmente ad ogni episodio, costringendo la politica a sviluppare robustezza rispetto all'incertezza del modello. Formalmente, si considera un insieme di parametri del simulatore $\xi \in \Xi$ (masse, attriti, rigidezze, inerzie, ritardi) distribuiti secondo $p(\xi)$. La politica ottimizzata è:

$$\pi^* = \arg\max_\pi \mathbb{E}_{\xi \sim p(\xi)} \mathbb{E}_{\tau \sim p(\tau \mid \pi, \xi)}[\Sigma_t \gamma^t r_t]$$

Essa massimizza il ritorno atteso rispetto sia alle traiettorie sia alla distribuzione dei parametri. La speranza è che una politica robusta rispetto a una sufficiente varietà di simulatori fisici si trasferisca al sistema reale, il cui ξ_{real} è come un campione della distribuzione $p(\xi)$ con buona probabilità.

La scelta della distribuzione $p(\xi)$ è critica: una distribuzione troppo ristretta non copre il vero simulatore, una troppo ampia rende il task irrisolvibile. La *adaptive domain randomization* stima on-line il parametro ξ_{real} tramite filtro bayesiano e restringe progressivamente la distribuzione di randomizzazione attorno alla stima corrente, coniugando esplorazione e sfruttamento nella calibrazione del simulatore.

6. Rappresentazioni latenti per la percezione robotica: codifiche spaziali e temporali

Un robot equipaggiato con sensori visivi (telecamere RGB, telecamere depth, sensori tattili) deve estrarre in tempo reale rappresentazioni compatte e azionabili dell'ambiente. Le architetture di codifica visiva per la robotica si distinguono per l'inductive bias spaziale che incorporano.

Le *convolutional neural network* (CNN) risentono della loro equivarianza traslazionale — appropriata per la percezione di texture e pattern locali — ma non di quella rispetto a trasformazioni geometriche più generali (rotazione, scala, prospettiva). I *Vision Transformer* (ViT), grazie all'attenzione globale, apprendono relazioni spaziali arbitrarie

tra regioni dell'immagine, ma mancano di equivarianza per costruzione (l'embedding posizionale rompe l'equivarianza traslazionale).

Per la robotica di manipolazione, le rappresentazioni di *keypoint* offrono un compromesso attraente: invece di codificare l'intera immagine in un vettore denso, il modello identifica un insieme di punti chiave nello spazio 3D (ad esempio, le estremità delle dita del robot, i vertici dell'oggetto da manipolare, i centri di massa stimati) e usa le loro coordinate come rappresentazione compatta dello stato. La formulazione di Kulkarni et al. (2019) e l'architettura KeyNet apprendono automaticamente questi punti chiave in modo non supervisionato, minimizzando la loss di ricostruzione attraverso una bottleneck spaziale differenziabile.

La codifica spazio-temporale per i robot mobili richiede anche di rappresentare la mappa dell'ambiente (*mapping*) e la localizzazione del robot in essa (*localization*) — il problema SLAM (*Simultaneous Localization And Mapping*). I metodi neurali per il SLAM visivo, come NeRF-SLAM e GO-SLAM, combinano reti neurali radianti (*Neural Radiance Fields*, NeRF) con ottimizzazione di posa incrementale, ottenendo mappe implicite differenziabili dell'ambiente che si prestano alla pianificazione nello spazio latente.

7. Apprendimento per imitazione e politiche visuomotorie

Accanto all'apprendimento per rinforzo model-based, l'apprendimento per imitazione (*imitation learning*, IL) è diventato il paradigma dominante per addestrare politiche robotiche complesse, specialmente in robotica di manipolazione. La ragione è pragmatica: raccogliere dimostrazioni umane è spesso molto più efficiente che definire funzioni di ricompensa adeguate e aspettare la convergenza del RL.

Il metodo fondamentale è la *behavioral cloning* (BC): si raccoglie un dataset $\mathcal{D} = \{(o_i, a_i^*)\}$ di coppie osservazione-azione dimostrate da un esperto, e si addestra una politica π_θ minimizzando la cross-entropy (o la MSE per azioni continue):

$$\mathcal{L}_{BC} = \mathbb{E}_{(o,a^*) \sim \mathcal{D}}[\| \pi_\theta(o) - a^* \|_2^2]$$

Il problema fondamentale della BC è la *distributional shift*: la politica apprende a imitare l'esperto sulla distribuzione di stati dell'esperto, ma durante l'esecuzione si trova in stati che l'esperto non ha mai visitato (a causa di errori di esecuzione inevitabili), per i quali la politica produce azioni errate che portano a stati ancora più lontani dalla distribuzione di addestramento — un fenomeno di cascata (*covariate shift*).

Il *DAgger* (Dataset Aggregation) di Ross, Gordon e Bagnell (2011) corregge iterativamente questa deriva: al passo n, si esegue la politica corrente π_n, si chiede all'esperto di fornire azioni corrette per gli stati visitati da π_n, e si aggiorna il dataset incorporando queste nuove dimostrazioni:

$\mathcal{D}_{n+1} = \mathcal{D}_n \cup \{(o, a^*_{expert}(o)) : o \sim \pi_n\}$
$\pi_{n+1} = \text{argmin}_\pi \mathbb{E}_{o \sim \mathcal{D}_{n+1}}[||\pi(o) - a^*||^2]$

Sotto ipotesi di regolarità, DAgger converge a una politica con regret $O(\sqrt{T})$ rispetto all'esperto, rispetto all'$O(T^2)$ della behavioral cloning pura.

Le politiche di diffusione (*diffusion policies*) di Chi e collaboratori (2023) hanno recentemente rivoluzionato il campo: invece di predire una singola azione, la politica genera una distribuzione di azioni attraverso un processo di denoising diffusivo condizionato sull'osservazione. La forward diffusion contamina progressivamente la sequenza di azioni a^* con rumore gaussiano:

$q(a^k \mid a^0) = \mathcal{N}(a^k; \sqrt{\bar{\alpha}_k} \cdot a^0, (1 - \bar{\alpha}_k) I)$

dove $\bar{\alpha}_k = \Pi_{i=1}^k (1 - \beta_i)$ e β_i è lo schedule di rumore. Il modello di score $\varepsilon_\theta(a^k, o, k)$ apprende a stimare il rumore aggiunto:

$\mathcal{L}_{diffusion} = \mathbb{E}_{k, a^0, \varepsilon}[||\varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_k} a^0 + \sqrt{(1-\bar{\alpha}_k)} \varepsilon, o, k)||_2^2]$

La generazione avviene attraverso il processo di reverse diffusion di DDPM (Denoising Diffusion Probabilistic Model), iterato da $k = K$ fino a $k = 0$. La diffusion policy supera la BC e i modelli GMM nel modellare distribuzioni di azioni multimodali — ad esempio, quando un task ammette più soluzioni ugualmente valide — e mostra una robustezza notevole rispetto alla variabilità delle osservazioni.

8. Fondamenta unificate: la prospettiva dei grandi modelli embodied

L'evoluzione più recente della robotica incarnata converge verso architetture unificate che combinano modelli del mondo, comprensione del linguaggio e pianificazione in un unico sistema. RT-2 (Robotics Transformer 2, Google DeepMind 2023) fine-tuna un Vision-Language Model (VLM) pre-addestrato su dati robotici, sfruttando la ricca comprensione del mondo fisico implicita nei modelli di linguaggio per generalizzare a task e oggetti mai visti in addestramento robotico.

OpenVLA (Open Vision-Language-Action) e π_0 (Pi-Zero) di Physical Intelligence (2024) adottano architetture ancora più ambiziose: un modello fondazionale con miliardi di parametri che integra, attraverso fine-tuning efficiente su dati di manipolazione, la comprensione visiva, la comprensione del linguaggio e il controllo motorio in una politica unica end-to-end. La sfida di questi modelli non è tecnologica quanto epistemica: come valutare se un sistema possiede genuina comprensione del mondo fisico, o si limita a un'interpolazione sofisticata tra comportamenti visti in addestramento?

La risposta più rigorosa a questa domanda passa attraverso il quadro della generalizzazione compositiva: un sistema comprende il mondo fisico se è in grado di ricombinare in modo corretto concetti, relazioni e azioni in configurazioni nuove, mai incontrate durante l'addestramento. Misure di generalizzazione compositiva come SCAN, COGS e, per la robotica, RoboSuite OOD test, costituiscono i benchmark più discriminativi per separare la comprensione dalla memorizzazione.

9. Convergenza dei paradigmi: verso modelli del mondo embodied unificati

Riassumendo il percorso di questa monografia, si può osservare una convergenza profonda tra quattro tradizioni che parevano inizialmente distinte: la teoria bayesiana della percezione (Friston, MacKay), i modelli generativi profondi per la previsione sequenziale (Dreamer, RSSM), l'apprendimento di rappresentazioni per predizione latente (JEPA, VICReg), e il controllo model-based per la robotica incarnata (MPC, MPPI, diffusion policy).

La convergenza si articola su tre assi. Il primo è architetturale: tutte e quattro le tradizioni convergono verso l'uso di encoder profondi che comprimono le percezioni in rappresentazioni latenti, di modelli di transizione che propagano queste rappresentazioni nel tempo, e di meccanismi di pianificazione che ottimizzano sequenze di azioni nello spazio latente. Il secondo è teorico: tutte condividono il framework variazionale dell'ELBO, declinato diversamente ma riconducibile alla stessa struttura matematica — massimizzazione dell'informazione mutua tra rappresentazioni, minimizzazione della divergenza tra prior e posterior, regolarizzazione dell'incertezza epistemica. Il terzo è epistemologico: tutte rigettano la visione di un agente passivo che registra il mondo, abbracciando invece quella di un agente attivo che costruisce, aggiorna e impiega modelli predittivi dell'ambiente per guidare il proprio comportamento — l'idea di Craik che aprì questa trattazione.

Il problema aperto più profondo rimane quello dell'efficienza del campionamento: gli esseri umani apprendono modelli del mondo straordinariamente accurati da quantità di esperienza modestissime rispetto a qualsiasi sistema artificiale attuale. La via verso questo regime di *few-shot world modeling* passa, secondo la convergenza delle opinioni nel campo, attraverso rappresentazioni gerarchiche e strutturate, prior forti sulla fisica del mondo, e la capacità di trasferire la conoscenza tra domini attraverso astrazioni condivise — tutte proprietà che la JEPA gerarchica di LeCun aspira a fornire, e che la robotica incarnata avrà il compito di incarnare nel senso più letterale del termine.

V. Teoria dell'apprendimento auto-supervisionato: informazione mutua, metodi contrastivi e il paesaggio dei SSL

1. Il problema dell'apprendimento delle rappresentazioni come inferenza statistica

L'apprendimento delle rappresentazioni (*representation learning*) è il problema di trovare una funzione f: $\mathcal{X} \to \mathcal{Z}$ tale che le rappresentazioni z = f(x) siano massimamente utili per un insieme di compiti a valle. La difficoltà fondamentale è che durante il pretraining l'insieme dei compiti a valle è ignoto, e si deve quindi apprendere una rappresentazione universalmente buona — o, più realisticamente, buona per la classe di compiti che si ritiene rilevante.

La formulazione statistica più generale del problema si esprime come l'apprendimento della mappa che preserva le strutture statistiche sufficienti dell'ingresso rispetto a qualche segnale predittivo. Questa formulazione contiene, come casi speciali, tutti i metodi SSL moderni: l'analisi delle componenti principali (PCA) che preserva la varianza, l'autoencoder che preserva la struttura di ricostruzione, i metodi contrastivi che preservano la coerenza tra viste aumentate, e la JEPA che preserva la predittività delle rappresentazioni latenti tra contesti.

Il quadro teorico unificante è offerto dalla teoria dell'informazione. Si consideri una variabile casuale X con distribuzione p(x) su $\mathcal{X}$. Una rappresentazione Z = f(X) è caratterizzata dalla distribuzione marginale p(z) e dalla distribuzione condizionale p(z | x) (nel caso deterministico: una massa di probabilità punto). L'informazione mutua tra X e Z è:

$$I(X; Z) = \mathbb{E}_{p(x,z)}[\log p(z \mid x) / p(z)]$$
$$= H(Z) - H(Z \mid X)$$
$$= H(X) - H(X \mid Z)$$

dove H denota l'entropia di Shannon. Per una funzione deterministica f, $H(Z \mid X) = 0$ e quindi $I(X; Z) = H(Z)$, che è massimizzata quando Z è distribuita uniformemente sull'ipersfera (per rappresentazioni a norma unitaria) o secondo una gaussiana isotropa (per rappresentazioni non vincolate). Questo spiega perché la regolarizzazione che promuove la distribuzione uniforme delle rappresentazioni — come il termine di varianza in VICReg o il momentum collapse detector in BYOL — ha effetti benefici: aumenta implicitamente H(Z) e quindi I(X; Z).

2. La prospettiva dell'informazione mutua nei metodi contrastivi

I metodi contrastivi come MoCo, SimCLR e InfoNCE massimizzano una stima dell'informazione mutua tra due viste aumentate x e x' della stessa osservazione. La connessione formale tra InfoNCE e informazione mutua è stata stabilita da van den Oord, Li e Vinyals (2018): per N campioni con un positivo e $N - 1$ negativi estratti da p(z), la loss InfoNCE è:

$\mathcal{L}_{InfoNCE} = -\mathbb{E}[\log \exp(f(x)^T f(x')) / (\exp(f(x)^T f(x')) + \Sigma_{j=1}^{N-1} \exp(f(x)^T f(x^-_j)))]$

Si può dimostrare che questa loss è un limite inferiore alla informazione mutua I(X; X'), con la stima che migliora al crescere di N:

$I(X; X') \geq \log N - \mathcal{L}_{InfoNCE}$

Questa disuguaglianza ha una conseguenza importante: per massimizzare efficacemente l'informazione mutua, è necessario un numero elevato di campioni negativi. SimCLR usa tipicamente batch di dimensione 4096–8192 per campionare negativi in-batch; MoCo mantiene una coda (memory bank) di 65536 rappresentazioni passate come negativi.

La scelta degli aumenti di dati (*data augmentations*) è cruciale e costituisce una forma di prior sulla struttura dell'invarianza desiderata. Se le augmentation trasformano x in x' attraverso una trasformazione $T \in \mathcal{T}$, la rappresentazione apprende a essere invariante rispetto al gruppo $\mathcal{T}$. Le augmentation tipiche — ritaglio casuale, jitter di colore, flipping orizzontale, blur gaussiano — inducono invarianze rispetto a traslazione parziale, colore, riflessione e frequenze spaziali alte. Questa scelta implicita del gruppo di invarianza è un inductive bias epistemologicamente forte che può essere tanto un vantaggio (se le invarianze coincidono con quelle rilevanti per i task a valle) quanto uno svantaggio (se i task a valle richiedono sensibilità proprio alle trasformazioni che vengono scartate).

3. Il collasso dimensionale e la dimensione intrinseca delle rappresentazioni

Uno dei fenomeni più sottili nell'apprendimento contrastivo è il *dimensional collapse*: la tendenza delle rappresentazioni a concentrarsi su un sottospazio di dimensione molto inferiore a quella dello spazio latente nominale, anche quando il collasso totale è prevenuto. Questo fenomeno è stato studiato da Hua e collaboratori (2021) attraverso l'analisi dello spettro della matrice di covarianza empirica delle rappresentazioni.

Sia $Z = [z_1, ..., z_N]^T \in \mathbb{R}^{N \times d}$ la matrice delle rappresentazioni su un batch di N campioni. La matrice di covarianza empirica è $\hat{C} = Z^T Z / N$ (assumendo Z centrata). Gli autovalori $\lambda_1 \geq \lambda_2 \geq ... \geq \lambda_d \geq 0$ di $\hat{C}$ caratterizzano la distribuzione della varianza lungo le direzioni principali. Il collasso dimensionale si manifesta come la concentrazione della varianza su un numero ridotto di autovalori, ovvero un rapido decadimento dello spettro:

$\text{rank}_\varepsilon(\hat{C}) = \min\{k : \Sigma_{i=1}^k \lambda_i / \Sigma_{i=1}^d \lambda_i \geq 1 - \varepsilon\}$

I metodi che mirano esplicitamente a uno spettro uniforme — come Spectral Decoupled Contrastive Learning (SDCLR) e i metodi di whitening — ottimizzano direttamente la condizione $\lambda_i \approx c$ per ogni i. VICReg, come già discusso, ottiene un effetto simile attraverso i termini di varianza e covarianza. Barlow Twins di Zbontar e collaboratori (2021) minimizza la differenza tra la matrice di cross-correlazione delle rappresentazioni di due rami e la matrice identità:

$$\mathcal{L}_{BT} = \Sigma_i (1 - C_{ii})^2 + \lambda \Sigma_i \Sigma_{j \neq i} C_{ij}^2$$

dove $C_{ij} = \Sigma_b z_b^i z_b^j / \sqrt{(\Sigma_b (z_b^i)^2 \cdot \Sigma_b (z_b^j)^2)}$ è la matrice di cross-correlazione normalizzata tra le rappresentazioni dei due rami su un batch b. Il primo termine spinge le rappresentazioni corrispondenti ad essere massimamente correlate (invarianza); il secondo spinge le dimensioni diverse ad essere incorrelate (ridondanza ridotta). Barlow Twins è equivalente a VICReg nel limite di batch infinito, come dimostrato analiticamente da Garrido e collaboratori (2022).

Un sviluppo recente di rilievo teorico per le strategie anti-collasso è la famiglia Rectified LpJEPA (Chang e collaboratori, 2024), che generalizza la distribuzione target delle rappresentazioni dalla gaussiana isotropa standard — implicita nella formulazione originale di I-JEPA e V-JEPA — a una famiglia più ampia denominata Generalized Gaussian Rettificata (RGG). Nella RGG, la densità delle rappresentazioni latenti è proporzionale a $\exp(-\|z\|^p / p)$ per un parametro di forma $p > 0$ che controlla esplicitamente la sparsità: per $p = 2$ si recupera la gaussiana isotropa (regime denso), per $p \to 1$ si ottiene una distribuzione laplaciana con picco più acuto all'origine e code più pesanti (regime sparso), e per $p \to 0$ si converge a distribuzioni in cui la massa è concentrata su superfici sferiche (regime ultrasparso). Questo parametro è ottimizzabile congiuntamente all'encoder durante il training e consente di adattare la geometria dello spazio latente alle proprietà statistiche del dominio senza modificare l'obiettivo di predizione. I-JEPA nella sua formulazione originale corrisponde al caso limite $p = 2$, che è il regime denso della famiglia RGG.

La motivazione teorica di Rectified LpJEPA è duplice. Dal punto di vista della massima entropia, ogni membro della famiglia RGG è la distribuzione a massima entropia soggetta al vincolo $\mathbb{E}[\|z\|^p] =$ costante, il che significa che la scelta della forma della distribuzione target è equivalente alla scelta del momento dell'ordine p da regolarizzare — una generalizzazione principiata del whitening implicito discusso per VICReg. Dal punto di vista biologico, il collegamento con i codici neurali sparsi e non negativi del sistema visivo dei mammiferi è diretto: i neuroni della corteccia visiva primaria codificano gli stimoli visivi attraverso pochi neuroni attivi simultaneamente su molti inattivi, un principio di efficienza della rappresentazione sensoriale noto come sparse coding (Olshausen e Field, 1996). Le rappresentazioni JEPA con $p < 2$ inducono questa struttura di sparsità in modo controllabile, combinando i vantaggi di efficienza del paradigma della predizione latente con il prior biologicamente motivato delle codifiche sparse.

4. Apprendimento auto-supervisionato mascherato: MAE e BEiT

L'apprendimento mascherato (*masked autoencoding*, MAE) è un'alternativa al contrastive learning che si ispira direttamente ai modelli linguistici mascherati (BERT). L'architettura MAE di He e collaboratori (2021) divide un'immagine in patch, maschera casualmente una frazione elevata (tipicamente 75%) di esse, e addestra un encoder-decoder a ricostruire i patch mascherati dai valori raw dei pixel.

Formalmente, sia l'immagine $x \in \mathbb{R}^{H \times W \times C}$ suddivisa in $N_p = HW/p^2$ patch di dimensione $p \times p$. Un sottoinsieme casuale $\mathcal{M} \subset \{1, ..., N_p\}$ di patch viene mascherato. L'encoder ViT processa solo i patch non mascherati $x_{[N_p] \setminus \mathcal{M}}$ e ne produce le rappresentazioni latenti. Un decoder leggero riceve le rappresentazioni dei patch non mascherati e i token di mask (vettori appresi condivisi) nelle posizioni mascherate, e predice i valori dei pixel nei patch mascherati:

$$\mathcal{L}_{MAE} = (1/|\mathcal{M}|) \sum_{i \in \mathcal{M}} ||x_i - \hat{x}_i||_2^2$$

dove $\hat{x}_i$ è la ricostruzione del patch i prodotta dal decoder. La normalizzazione dei pixel all'interno di ogni patch (patch normalization) migliora significativamente le prestazioni, suggerendo che l'encoder deve imparare a predire le strutture relative (contrasti, bordi) piuttosto che i valori assoluti.

BEiT (BERT Pre-Training of Image Transformers) di Bao e collaboratori (2021) adotta una strategia di supervisione discreta: invece di predire i valori raw dei pixel, il modello predice i token visivi discreti prodotti da un tokenizer DALL-E (un VQVAE pre-addestrato). La loss è la cross-entropy sui token:

$$\mathcal{L}_{BEiT} = -(1/|\mathcal{M}|) \sum_{i \in \mathcal{M}} \sum_{v=1}^{|\mathcal{V}|} \mathbb{1}[v_i = v] \cdot \log p_\theta(v \mid x_{[N_p] \setminus \mathcal{M}}, pos_i)$$

dove $\mathcal{V}$ è il vocabolario dei token visivi e v_i è il token che corrisponde al patch i dell'immagine originale (non mascherata) secondo il tokenizer. La supervisione discreta agisce come una forma implicita di compressione dell'informazione: il modello deve predire solo la categoria semantica di ogni patch, non i suoi dettagli pixelici, il che promuove rappresentazioni più semantiche.

La distinzione tra MAE (predizione nello spazio degli input) e I-JEPA (predizione nello spazio latente) diventa qui critica. MAE ricostruisce i pixel mascherati e quindi deve allocare capacità alla modellazione dei dettagli texturali; I-JEPA predice le rappresentazioni latenti del ramo target e può ignorare i dettagli irrilevanti per la struttura latente. Gli esperimenti di Assran e collaboratori mostrano che I-JEPA produce rappresentazioni con maggiore qualità semantica a parità di budget computazionale, confermando l'intuizione di LeCun sulla superiorità della predizione nello spazio latente.

5. Geometria delle rappresentazioni e struttura iperbolica

La scelta dello spazio latente — euclideo, iperbolico, sferico, prodotto di spazi — ha implicazioni profonde per la qualità delle rappresentazioni apprese. Il mondo fisico e biologico presenta strutture gerarchiche ubique: tassonomie biologiche, strutture sintattiche, relazioni parte-intero negli oggetti fisici. Lo spazio euclideo è inadatto a rappresentare gerarchie profonde: per embedare un albero con n nodi con distorsione ε, è necessaria una dimensione $\Omega(\log n / \varepsilon^2)$ nello spazio euclideo, mentre lo spazio iperbolico con curvatura negativa $K < 0$ può farlo in dimensione $O(\log n)$.

Lo spazio iperbolico di Poincaré in dimensione d è il disco aperto $D^d = \{x \in \mathbb{R}^d : ||x||_2 < 1\}$ dotato della metrica riemanniana:

$$ds^2 = 4 / (1 - ||x||^2)^2 \cdot \Sigma_i \, dx_i^2$$

La distanza geodetica tra due punti $x, y \in D^d$ è:

$$d_H(x, y) = \operatorname{arccosh}(1 + 2||x - y||^2 / ((1 - ||x||^2)(1 - ||y||^2)))$$

Le reti neurali iperboliche (Hyperbolic Neural Networks, Ganea et al. 2018) operano in questo spazio attraverso operazioni gyrovettoriali. La somma di Möbius generalizza l'addizione euclidea:

$$x \oplus_c y = ((1 + 2c\langle x,y\rangle + c||y||^2)x + (1 - c||x||^2)y) / (1 + 2c\langle x,y\rangle + c^2||x||^2||y||^2)$$

dove $c = -K > 0$ è la curvatura assoluta. La moltiplicazione scalare di Möbius è:

$$r \otimes_c x = (1/\sqrt{c}) \tanh(r \cdot \operatorname{arctanh}(\sqrt{c}\,||x||)) \cdot x / ||x||$$

Le reti neurali iperboliche sostituiscono le operazioni lineari euclidee con le loro controparti gyrovettoriali e trasportano le funzioni di attivazione sulla mappa esponenziale/logaritmica tra spazio tangente (euclideo) e varietà iperbolica. L'applicazione di questi spazi ai world model è ancora allo stato nascente, ma risultati preliminari suggeriscono che rappresentazioni iperboliche delle gerarchie di oggetti migliorano la generalizzazione compositiva nelle task di manipolazione.

6. Il problema della predittività e la statistica sufficiente per il futuro

Una formulazione rigorosa del problema dell'apprendimento auto-supervisionato per i modelli del mondo deve affrontare la domanda: quale struttura deve avere la rappresentazione appresa perché sia massimamente utile per la previsione del futuro?

La risposta formale è data dal concetto di *predictive sufficient statistic*. Sia X_{1:t} la storia dell'osservazione fino al tempo t e Y_{t+1:t+H} l'osservazione futura sull'orizzonte H. Una statistica S_t = S(X_{1:t}) è sufficiente per Y_{t+1:t+H} rispetto a X_{1:t} se e solo se:

$$I(X_{1:t}; Y_{t+1:t+H} \mid S_t) = 0$$

ovvero, condizionato su S_t, la storia passata non fornisce informazione aggiuntiva sul futuro. Il *predictive information* di Bialek e collaboratori misura la quantità totale di informazione condivisa tra passato e futuro:

$$I_{pred} = I(X_{past}; X_{future}) = H(X_{future}) - H(X_{future} \mid X_{past})$$

Nei sistemi ergodici stazionari, la predictive information può essere decomposita sulle scale temporali τ:

$$I_{pred} = \int_0^\infty i(\tau)\, d\tau, \quad i(\tau) = -d/d\tau\, [H(X_{t+\tau} \mid X_{1:t})]$$

Un buon encoder per i world model dovrebbe catturare le scale temporali τ di predictive information rilevanti per i task di controllo, che tipicamente corrispondono agli orizzonti delle azioni pianificate. Questa prospettiva giustifica la scelta di orizzonti di masking diversi in V-JEPA e suggerisce che un masking multi-scala (tube di lunghezze diverse) potrebbe catturare una gamma più ampia di strutture predittive.

7. Metodi di distillazione della conoscenza come SSL implicito

La distillazione della conoscenza (*knowledge distillation*, KD) trasferisce le rappresentazioni di un modello insegnante di grandi dimensioni a uno studente più piccolo. Ma i metodi di distillazione feature-level — come FitNets, PKT e le recenti varianti self-distillation — possono essere interpretati come metodi SSL in cui il "dato aumentato" è la rappresentazione dell'insegnante su un diverso esempio della distribuzione.

DINO (Self-DIstillation with NO labels, Caron et al. 2021) è il caso più influente di self-distillation come SSL: uno studente f_θ è addestrato a imitare un insegnante g_ψ (aggiornato con EMA) su viste diverse della stessa immagine. La loss è la cross-entropy tra la distribuzione di probabilità dello studente e quella (sharpened) dell'insegnante su K_local viste locali (crops piccoli) e K_global viste globali:

$$\mathcal{L}_{DINO} = -\Sigma_{x \in x_{global}} \Sigma_{x' \in x_{all} \setminus \{x\}} p_\psi(x)^T \log p_\theta(x')$$

dove $p_\theta(x) = \text{softmax}(f_\theta(x) / \tau_s)$ e $p_\psi(x) = \text{softmax}(g_\psi(x) / \tau_t)$ con $\tau_t < \tau_s$ (temperatura dell'insegnante più bassa, distribuzione più appuntita). Il centering dell'uscita dell'insegnante — sottrazione di una media esponenziale mobile — previene il collasso verso un singolo token dominante:

$$g_\psi(x) \leftarrow g_\psi(x) - c, \quad c \leftarrow m_c \cdot c + (1 - m_c) \cdot (1/B) \Sigma_b g_\psi(x_b)$$

DINO v2 (Oquab et al. 2023) scala questo approccio a 142M parametri e addestramenti su un dataset curato di 142M immagini, raggiungendo rappresentazioni che si avvicinano all'ottimo per un ampio spettro di benchmark visivi. L'analisi delle mappe di attenzione di DINOv2 rivela emergenza spontanea di segmentazione semantica: le teste di attenzione imparano a raggruppare pixel dello stesso oggetto senza alcuna supervisione semantica, un comportamento che costituisce evidenza indiretta che il modello ha costruito un modello implicito della struttura oggettuale del mondo visivo.

8. Apprendimento contrastivo temporale e Contrastive Predictive Coding

Il *Contrastive Predictive Coding* (CPC) di van den Oord e collaboratori (2018) è il precursore dei metodi contrastivi temporali e ha ispirato direttamente la JEPA. L'idea fondamentale è di apprendere rappresentazioni che siano predittive del futuro in un senso contrastivo: la rappresentazione del contesto c_t deve essere simile alle rappresentazioni degli osservazioni future z_{t+k} (positivi) e dissimile dalle rappresentazioni di osservazioni non correlate (negativi).

Il modello CPC si compone di un encoder g_{enc} che mappa $x_t \to z_t$ e di un modello autoregressivo g_{ar} che aggrega $z_{1:t}$ in un contesto c_t. Un predittore lineare W_k predice la rappresentazione k passi nel futuro:

$$\hat{z}_{t+k} = W_k \cdot c_t$$

La loss NCE (Noise Contrastive Estimation) per ogni orizzonte k è:

$$\mathcal{L}_k = -\mathbb{E}[\log (\exp(\hat{z}_{t+k}^T z_{t+k}) / \Sigma_{j=0}^{N} \exp(\hat{z}_{t+k}^T z_j))]$$

dove $N - 1$ negativi $\{z_j\}_{j=1}^{N-1}$ *sono campionati dalla stessa sequenza (altri timestep) o da sequenze diverse del batch. Si può dimostrare che minimizzare questa loss massimizza un limite inferiore alla informazione mutua* $I(c_t; x_{t+k})$:

$$I(c_t; x_{t+k}) \geq \log N - \mathcal{L}_k$$

CPC è stato applicato con successo all'audio (speech), al testo, alle immagini e ai video, e costituisce la dimostrazione teoricamente più fondata che i metodi contrastivi temporali apprendono strutture predittive significative. La differenza rispetto alla JEPA è che CPC usa predittori lineari e loss contrastiva, mentre la JEPA usa predittori più profondi e loss di regressione nello spazio latente; entrambe convergono sull'idea di predizione nello spazio delle rappresentazioni.

VI. Modelli del mondo in tempo continuo: Neural ODE, reti hamiltoniane e lagrangiane, integrazione simplettica

1. Il limite continuo dei modelli di transizione discreti

Le architetture discusse nelle parti precedenti — RSSM, LSTM, GRU — modellano la dinamica del mondo come una successione discreta di transizioni di stato a intervalli temporali fissi. Questa scelta è computazionalmente conveniente ma introduce una dipendenza arbitraria dal passo di campionamento: un modello addestrato con osservazioni a 30 Hz non si generalizza direttamente a sequenze campionate a frequenza diversa, e non cattura la struttura continua della dinamica sottostante.

Il framework delle *Neural Ordinary Differential Equations* (Neural ODE), introdotto da Chen e collaboratori nel 2018, propone di parametrizzare la derivata temporale dello stato latente con una rete neurale, trattando l'evoluzione dello stato come la soluzione di un sistema di equazioni differenziali ordinarie:

$$dz/dt = f_\theta(z(t), t)$$

dove $z(t) \in \mathbb{R}^d$ è lo stato latente all'istante t e $f_\theta: \mathbb{R}^d \times \mathbb{R} \to \mathbb{R}^d$ è una rete neurale profonda. Dato uno stato iniziale $z(t_0) = z_0$, la soluzione al tempo t_1 si ottiene integrando numericamente:

$$z(t_1) = z(t_0) + \int_{t_0}^{t_1} f_\theta(z(t), t)\, dt = \mathrm{ODESolve}(f_\theta, z_0, t_0, t_1)$$

L'addestramento richiede il calcolo del gradiente della loss $\mathcal{L}$ rispetto ai parametri θ attraverso l'integratore ODE. Il metodo diretto — backpropagation through the ODE solver — è computazionalmente dispendioso e numericamente instabile per orizzonti lunghi. Il metodo del *metodo aggiunto* (*adjoint method*), riscoperto da Pontryagin e generalizzato da Chen al caso dei Neural ODE, risolve questo problema in modo elegante.

2. Il metodo aggiunto: derivazione completa

Si definisce lo stato aggiunto $a(t) = \partial \mathcal{L} / \partial z(t)$ come la sensibilità della loss rispetto allo stato al tempo t. L'evoluzione di a(t) è governata da un'ODE aggiunta (*adjoint ODE*) che si integra all'indietro nel tempo da t_1 a t_0:

$$da/dt = -a(t)^T \cdot \partial f_\theta(z(t), t)/\partial z$$

Con condizione al contorno $a(t_1) = \partial \mathcal{L} / \partial z(t_1)$. Il gradiente della loss rispetto ai parametri θ si ottiene integrando simultaneamente lo stato aggiunto:

$$d\mathcal{L}/d\theta = -\int_{t_1}^{t_0} a(t)^T \cdot \partial f_\theta(z(t), t)/\partial \theta \, dt$$

Questa derivazione può essere ottenuta applicando il teorema di Lagrange ai moltiplicatori della ODE vincolata:

$$\min_\theta \mathcal{L}(z(t_1)) \quad \text{soggetto a:} \quad dz/dt = f_\theta(z, t), \quad z(t_0) = z_0$$

Si introduce la funzione hamiltoniana aggiunta:

$$H(z, a, \theta, t) = a^T f_\theta(z, t)$$

Le equazioni di Hamilton-Pontryagin danno:

$$dz/dt = +\partial H/\partial a = f_\theta(z, t)$$
$$da/dt = -\partial H/\partial z = -a^T (\partial f_\theta/\partial z)$$

che è esattamente l'ODE aggiunta. Il gradiente rispetto ai parametri è:

$$d\mathcal{L}/d\theta = \int_{t_0}^{t_1} \partial H/\partial \theta \, dt = \int_{t_0}^{t_1} a(t)^T \partial f_\theta(z,t)/\partial \theta \, dt$$

Il vantaggio cruciale del metodo aggiunto è che richiede memoria O(1) rispetto all'integrazione (solo lo stato corrente z(t) e lo stato aggiunto a(t) devono essere mantenuti), a differenza della backpropagation diretta che richiede O(N_steps) memoria per memorizzare tutti gli stati intermedi.

3. Latent Neural ODEs per le serie temporali irregolari

Un'applicazione particolarmente rilevante delle Neural ODE è la modellazione di serie temporali osservate a istanti irregolari — frequente nella robotica (sensori a rate

variabile), nella medicina (visite cliniche irregolari) e nei sistemi fisici (campionamento adattivo). L'architettura Latent ODE di Rubanova e collaboratori (2019) combina un encoder RNN bidirezionale con una Neural ODE nello spazio latente.

Dato un insieme di osservazioni irregolari $\{(x_i, t_i)\}_{i=1}^N$ con $t_0 < t_1 < ... < t_N$, il modello:

Prima fase — codifica: un encoder RNN processa le osservazioni in ordine inverso (da t_N a t_0) e produce la distribuzione variazionale sullo stato iniziale:

$$q_\varphi(z_0 \mid x_{1:N}, t_{1:N}) = \mathcal{N}(\mu_0, \sigma_0^2)$$

con $(\mu_0, \sigma_0) = \text{RNN}_\varphi(x_N, ..., x_1)$.

Seconda fase — propagazione: lo stato latente z_0 viene propagato in avanti attraverso la Neural ODE per produrre $z(t)$ ad ogni istante di osservazione desiderato:

$$z(t_i) = \text{ODESolve}(f_\theta, z(t_{i-1}), t_{i-1}, t_i)$$

Terza fase — decodifica: ogni stato $z(t_i)$ viene decodificato in $\hat{x}_i$ attraverso un decoder $p_\theta(x_i \mid z(t_i))$.

La loss è l'ELBO standard del VAE, con la particolarità che il termine di ricostruzione è definito solo sugli istanti di osservazione disponibili:

$$\mathcal{L} = \Sigma_{i=1}^N \mathbb{E}_q[\log p_\theta(x_i \mid z(t_i))] - \text{KL}[q_\varphi(z_0 \mid x, t) \,||\, p(z_0)]$$

Questo framework si generalizza naturalmente al caso di osservazioni multivariate con pattern di missingness arbitrari, aprendo la strada a modelli del mondo in grado di operare su dati sensoriali eterogenei e irregolari — condizione comune in robotica reale.

4. Reti hamiltoniane: struttura simplettica e conservazione dell'energia

Un limite fondamentale delle Neural ODE generiche è che la rete f_θ non è vincolata a rispettare le leggi fisiche di conservazione. Per sistemi meccanici — che costituiscono la grande maggioranza dei robot fisici — la dinamica è governata dalle equazioni di Hamilton, che preservano l'energia totale e lo spazio delle fasi simplettico. Le *Hamiltonian Neural Networks* (HNN) di Greydanus e collaboratori (2019) incorporano questa struttura come inductive bias architetturale.

Un sistema hamiltoniano in dimensione $2n$ è descritto dallo spazio delle fasi $(q, p) \in \mathbb{R}^n \times \mathbb{R}^n$ (coordinate generalizzate e momenti coniugati) e da una funzione

hamiltoniana $\mathcal{H}$: $\mathbb{R}^{2n} \to \mathbb{R}$ che rappresenta l'energia totale del sistema. Le equazioni del moto sono:

$$dq_i/dt = +\partial\mathcal{H}/\partial p_i$$
$$dp_i/dt = -\partial\mathcal{H}/\partial q_i$$

In forma compatta, definendo lo stato $y = (q, p)^T \in \mathbb{R}^{2n}$ e la matrice simplettica $J = [[0, I_n], [-I_n, 0]] \in \mathbb{R}^{2n \times 2n}$:

$$dy/dt = J \cdot \nabla_y \mathcal{H}(y)$$

Una HNN parametrizza $\mathcal{H}$ con una rete neurale $\mathcal{H}_\theta: \mathbb{R}^{2n} \to \mathbb{R}$ e definisce il campo vettoriale della dinamica come:

$$\dot{y} = J \cdot \nabla_y \mathcal{H}_\theta(y)$$

Il gradiente $\nabla_y \mathcal{H}_\theta$ si calcola tramite differenziazione automatica attraverso la rete $\mathcal{H}_\theta$. L'addestramento minimizza la discrepanza tra il campo vettoriale predetto e quello osservato:

$$\mathcal{L}_{HNN} = ||J \cdot \nabla_y \mathcal{H}_\theta(y) - \dot{y}_{obs}||_2^2$$

dove $\dot{y}_{obs}$ sono le derivate temporali osservate (stimate numericamente dai dati). Per costruzione, la dinamica definita da una HNN conserva esattamente l'energia: $d\mathcal{H}_\theta/dt = \nabla_y \mathcal{H}_\theta^T \dot{y} = \nabla_y \mathcal{H}_\theta^T J \nabla_y \mathcal{H}_\theta = 0$, poiché J è antisimmetrica e il prodotto ($v^T J v = 0$ per ogni v). Questa proprietà vale esattamente nel modello continuo, ma viene violata nell'integrazione numerica discreta — problema che discuteremo nella sezione sull'integrazione simplettica.

5. Reti lagrangiane e la formulazione di Euler-Lagrange neurale

Un'alternativa alle HNN che si rivela più naturale per i robot a giunto è offerta dalle *Lagrangian Neural Networks* (LNN) di Cranmer, Greydanus e Kochkov (2020). La formulazione lagrangiana opera nello spazio delle configurazioni $(q, \dot{q}) \in \mathbb{R}^n \times \mathbb{R}^n$ e parametrizza il lagrangiano $\mathcal{L}_\theta(q, \dot{q}) = K_\theta(q, \dot{q}) - V_\theta(q)$, dove K è l'energia cinetica e V il potenziale.

Le equazioni di Euler-Lagrange danno le equazioni del moto:

$$d/dt\,(\partial\mathcal{L}/\partial\dot{q}_i) - \partial\mathcal{L}/\partial q_i = \tau_i$$

Sviluppando la derivata totale:

$$\Sigma_j \left(\partial^2 \mathcal{L}/\partial \dot{q}_i \partial \dot{q}_j\right) \ddot{q}_j + \Sigma_j \left(\partial^2 \mathcal{L}/\partial \dot{q}_i \partial q_j\right) \dot{q}_j - \partial \mathcal{L}/\partial q_i = \tau_i$$

In forma matriciale:

$$M_\theta(q)\, \ddot{q} = \tau - C_\theta(q, \dot{q})\, \dot{q} + g_\theta(q)$$

dove $M_\theta = \partial^2 K_\theta/\partial \dot{q}^2$, C_θ e g_θ si derivano analiticamente da $\mathcal{L}_\theta$ attraverso le relazioni di Euler-Lagrange. Una LNN che modella il lagrangiano come:

$$\mathcal{L}_\theta(q, \dot{q}) = \tfrac{1}{2}\, \dot{q}^\top M_\theta(q)\, \dot{q} - V_\theta(q)$$

garantisce per costruzione che M_θ sia simmetrica. Imponendo $M_\theta = L_\theta(q)\, L_\theta(q)^\top + \varepsilon I$ (decomposizione di Cholesky neurale) si ottiene anche la definite positività, rispettando la fisica della matrice di inerzia.

La differenza fondamentale tra LNN e HNN per la robotica è che le LNN operano naturalmente in coordinata q (angoli dei giunti), mentre le HNN richiedono la trasformazione allo spazio dei momenti $\dot{q} \to p = \partial \mathcal{L}/\partial \dot{q}$, che per sistemi con metrica non piatta ($\dot{q} \to p$ non è una semplice moltiplicazione per una costante) può introdurre complessità aggiuntive. Per i robot seriali, le LNN risultano generalmente più stabili e interpretabili.

6. Integrazione simplettica: l'algoritmo di Störmer-Verlet e il metodo di Ruth-Forest

Le equazioni hamiltoniane (e lagrangiane) definiscono un flusso simplettico: la mappa di evoluzione Φ_t: $(q(0), p(0)) \to (q(t), p(t))$ preserva la 2-forma simplettica $\omega = \Sigma_i\, dp_i \wedge dq_i$. Questa proprietà implica la conservazione del volume nello spazio delle fasi (teorema di Liouville) e delle quantità conservate del sistema.

Gli integratori numerici classici (Eulero esplicito, Runge-Kutta di ordine 4) non preservano la struttura simplettica e introducono una deriva sistematica nell'energia totale proporzionale a Δt^2 (RK4) o Δt (Eulero). Gli integratori simplettici, costruiti in modo da preservare esattamente ω a ogni passo, mostrano invece un'oscillazione dell'energia intorno al valore corretto senza deriva sistematica, rendendoli ideali per simulazioni a lungo termine.

L'algoritmo di Störmer-Verlet (o leapfrog) per un sistema hamiltoniano separabile $\mathcal{H}(q, p) = K(p) + V(q)$:

$$p_{n+1/2} = p_n - (\Delta t/2)\, \nabla_q V(q_n)$$
$$q_{n+1} = q_n + \Delta t\, \nabla_p K(p_{n+1/2})$$

$$p_{n+1} = p_{n+1/2} - (\Delta t/2)\, \nabla_q V(q_{n+1})$$

Si può dimostrare che questo schema corrisponde all'integrazione esatta di un hamiltoniano modificato $\tilde{\mathcal{H}} = \mathcal{H} + O(\Delta t^2)$, e che la mappa $(q_n, p_n) \to (q_{n+1}, p_{n+1})$ è simplettica per costruzione. La conservazione dell'energia oscilla ma rimane $O(\Delta t^2)$-vicina all'energia originale per tempi arbitrariamente lunghi.

Per le Neural ODE hamiltoniane, la scelta dell'integratore simplettico durante il training e l'inferenza è cruciale. Sanchez-Gonzalez e collaboratori (2019) e Chen e collaboratori (2020) mostrano che addestrare una HNN con integratore simplettico (invece di RK4) produce modelli con errori di energia molto inferiori su orizzonti lunghi, specialmente per dinamiche conservative come il pendolo invertito o i sistemi multilink.

L'integrazione simplettica per sistemi hamiltoniani non separabili — dove $\mathcal{H}(q, p)$ non si fattorizza in $K(p) + V(q)$ — richiede metodi più sofisticati, come il metodo di Ruth-Forest o gli integratori di composizione di Yoshida. Il metodo di Yoshida di ordine 4 compone tre passi Störmer-Verlet con coefficienti:

$$w_1 = 1 / (2 - 2^{1/3}), \quad w_0 = -2^{1/3} / (2 - 2^{1/3}),$$
$$c_1 = c_4 = w_1/2, \; c_2 = c_3 = (w_0 + w_1)/2, \; d_1 = d_3 = w_1, \; d_2 = w_0$$

producendo un integratore simplettico di ordine 4 con 4 valutazioni del gradiente per passo.

7. Equazioni differenziali stocastiche e la dinamica con rumore

I sistemi fisici reali non sono mai perfettamente deterministici: il rumore termico, le perturbazioni esterne e l'incertezza di misura rendono la dinamica intrinsecamente stocastica. La generalizzazione naturale delle Neural ODE al caso stocastico sono le *Neural Stochastic Differential Equations* (Neural SDE), che estendono il framework di Itō:

$$dz = f_\theta(z, t)\, dt + g_\varphi(z, t)\, dW_t$$

dove W_t è un moto browniano standard su $\mathbb{R}^d$, f_θ è il drift neurale e g_φ è la diffusione neurale. L'addestramento di Neural SDE richiede l'estensione del metodo aggiunto al caso stocastico, derivata da Li e collaboratori (2020):

$$da = -[\partial f_\theta/\partial z^T a + \partial g_\varphi/\partial z^T v]\, dt + v \cdot dW_t$$

dove W_t è un moto browniano retrograde e $v(t)$ è un processo aggiunto stocastico. La complessità computazionale di questo training è significativamente superiore al

caso deterministico e richiede l'uso di discretizzazioni stocastiche efficienti come Euler-Maruyama o Milstein.

Le Neural SDE sono strettamente legate ai modelli di diffusione: una SDE con drift di score function $\nabla_z \log p_t(z)$ e diffusione costante σ recupera esattamente il processo di diffusione inversa di Song e collaboratori (2021). Questa connessione unifica i modelli di diffusione, le Neural ODE e gli EBM sotto un unico framework differenziale stocastico, con implicazioni profonde per la modellazione generativa di traiettorie robotiche.

8. Equivariant Neural ODE e simmetrie fisiche

Un'ulteriore raffinamento delle Neural ODE per la fisica riguarda l'equivarianza rispetto ai gruppi di simmetria del sistema. Per un robot che opera nello spazio 3D, le leggi fisiche sono equivarianti rispetto al gruppo di Euclide $E(3) = O(3) \ltimes \mathbb{R}^3$ (rotazioni, riflessioni e traslazioni). Un modello di transizione che rispetta questa simmetria generalizza automaticamente tra configurazioni spaziali diverse dello stesso scenario fisico.

Le *Equivariant Graph Neural ODE* parametrizzano il campo vettoriale come:

$$dz_i/dt = f_\theta(\{z_j, x_j - x_i\}_j)$$

dove $z_i \in \mathbb{R}^d$ è la feature interna del nodo i (invariante rispetto a E(3)) e $x_i \in \mathbb{R}^3$ è la posizione spaziale del nodo (equivariante). La funzione f_θ è costruita in modo che trasformi in modo covariante rispetto a E(3): se si applica una trasformazione euclidea (R, t) a tutte le posizioni, le velocità prodotte ruotano anch'esse di R e le feature interne rimangono invariate.

L'architettura EGNN (Equivariant Graph Neural Network, Satorras et al. 2021) realizza questa proprietà aggiornando le posizioni e le feature congiuntamente:

$$m_{ij} = \varphi_e(z_i, z_j, ||x_i - x_j||_2^2, a_{ij})$$
$$x_i \leftarrow x_i + \Sigma_{j \neq i} (x_i - x_j) \varphi_x(m_{ij})$$
$$z_i \leftarrow \varphi_h(z_i, \Sigma_{j \neq i} m_{ij})$$

dove φ_e, φ_x, φ_h sono MLP e a_{ij} sono attributi aggiuntivi degli archi. Si verifica facilmente che l'aggiornamento delle posizioni è equivariante rispetto a traslazioni e rotazioni, mentre le feature z_i sono invarianti.

L'applicazione delle EGNN come modelli di transizione in RSSM per sistemi molecolari e robotici multi-corpo mostra generalizzazioni superiori a modelli non equivarianti, specialmente per configurazioni di test fuori distribuzione che coinvolgono orientazioni o posizioni non viste durante l'addestramento.

VII. Pianificazione sotto incertezza: POMDP, spazio delle credenze e inferenza attiva

1. Il framework POMDP: definizione assiomatica

Il Processo Decisionale di Markov Parzialmente Osservabile (*Partially Observable Markov Decision Process*, POMDP) è il formalismo matematico più generale per la pianificazione sequenziale in ambienti con incertezza sullo stato. Un POMDP è definito dalla settupla ($\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$, T, R, Z, γ) dove:

$\mathcal{S}$ è lo spazio degli stati (lo stato "vero" del mondo, non direttamente accessibile all'agente), $\mathcal{A}$ è lo spazio delle azioni, $\mathcal{O}$ è lo spazio delle osservazioni. T: $\mathcal{S} \times \mathcal{A} \times \mathcal{S} \rightarrow [0,1]$ è la funzione di transizione T(s, a, s') = p(s' | s, a). R: $\mathcal{S} \times \mathcal{A} \rightarrow \mathbb{R}$ è la funzione di ricompensa. Z: $\mathcal{S} \times \mathcal{A} \times \mathcal{O} \rightarrow [0,1]$ è la funzione di osservazione Z(s', a, o) = p(o | s', a). $\gamma \in [0,1)$ è il fattore di sconto.

Poiché l'agente non osserva s direttamente, mantiene una *belief state* b: $\mathcal{S} \rightarrow [0,1]$ — una distribuzione di probabilità sullo spazio degli stati — che riassume tutta l'informazione disponibile sulla storia passata in modo sufficiente per la decisione ottimale. Si dimostra (Theorem of Sufficient Statistics per POMDP) che b è una statistica sufficiente per la politica ottimale, ovvero $\pi^*(a \mid b)$ è ottimale sapendo solo b e non richiedendo l'intera storia ($o_{1:t}$, $a_{1:t-1}$).

L'aggiornamento della belief state dopo aver eseguito l'azione a e aver ricevuto l'osservazione o è dato dalla regola di Bayes per POMDP:

$$b'(s') = \eta \cdot Z(s', a, o) \cdot \Sigma_{s \in \mathcal{S}} T(s, a, s') \cdot b(s)$$

dove η è la costante di normalizzazione $\eta = 1 / \Sigma_{s'} Z(s', a, o) \Sigma_s T(s, a, s') b(s)$. Questo aggiornamento, denominato *belief update*, definisce la funzione di transizione nello spazio delle belief:

$$\tau: \Delta(\mathcal{S}) \times \mathcal{A} \times \mathcal{O} \rightarrow \Delta(\mathcal{S})$$
$$b' = \tau(b, a, o)$$

Lo spazio delle belief $\Delta(\mathcal{S})$ è un simplesso di dimensione $|\mathcal{S}| - 1$ e costituisce, esso stesso, un MDP completamente osservabile (il *belief MDP*) con stati b, azioni a, transizioni τ e ricompensa $r(b, a) = \Sigma_s b(s) R(s, a)$. La soluzione del POMDP coincide con la soluzione del belief MDP.

2. La funzione di valore per POMDP: rappresentazione per iperpiani

La funzione di valore ottimale del belief MDP V*(b) è convessa in b (essendo un massimo di funzioni affini) e può essere rappresentata come il massimo di un insieme finito di vettori $\alpha \in \mathbb{R}^{|\mathcal{S}|}$ (*alpha-vectors*):

$$V^*(b) = \max_{\alpha \in \Gamma} \alpha^T b$$

dove Γ è un insieme di vettori α (potenzialmente di cardinalità esponenziale nel numero di stati). Ogni vettore α corrisponde a una politica condizionale (*conditional plan*) che prescrive quale azione intraprendere a ogni nodo dell'albero di decisione condizionato alle osservazioni ricevute.

L'operatore di backup (*backup operator*) del POMDP, che trasforma un insieme di vettori alpha al passo t+1 in quelli al passo t, è:

$$\alpha_{a,o}^{\alpha'}(s) = Z(s, a, o) \Sigma_{s'} T(s, a, s') \alpha'(s')$$

$$\alpha_a^{\alpha'_{1:O}}(s) = R(s, a) + \gamma \Sigma_o \max_{j} \alpha_{a,o}^{\alpha'_j}(s)$$

L'operatore di backup produce per ogni azione a e per ogni combinazione di vettori alpha sull'insieme delle osservazioni un nuovo vettore alpha-vector. Il numero di vettori generati cresce esponenzialmente nel numero di osservazioni per ogni passo di backup, rendendo la soluzione esatta intrattabile per problemi di dimensione realistica.

Gli algoritmi approssimati come PBVI (*Point-Based Value Iteration*), SARSOP e POMCP risolvono questo problema restringendo il backup a un insieme finito di punti rappresentativi nello spazio delle belief, campionati attraverso simulazioni forward della politica corrente.

3. Incertezza epistemica e aletoristica: distinzione formale

Nel contesto dei modelli del mondo neurali, è fondamentale distinguere tra due categorie concettualmente distinte di incertezza:

L'*incertezza epistemica* (*epistemic uncertainty*, EU) è dovuta alla mancanza di dati: il modello non ha visto abbastanza esempi da una regione dello spazio degli stati per stimarne accuratamente la dinamica. Questa incertezza può, in linea di principio, essere ridotta raccogliendo più dati. Si manifesta come variabilità nelle previsioni tra modelli diversi addestrati sullo stesso dataset (varianza dell'ensemble) o come entropia nella distribuzione posteriore bayesiana sui parametri del modello.

L'*incertezza aleatoria* (*aleatoric uncertainty*, AU) è intrinseca al sistema: anche con un modello perfettamente identificato, la dinamica del mondo è stocasticamente incerta a causa di rumore irriducibile, caos deterministico o parziale osservabilità. Non può

essere ridotta con più dati; può solo essere modellata accuratamente. Si manifesta come varianza del modello di transizione stocastica condizionata allo stato.

Formalmente, per un modello di transizione p(s_{t+1} | s_t, a_t; θ) con incertezza bayesiana sui parametri p(θ | $\mathcal{D}$):

$$\mathrm{Var}[s_{t+1} \mid s_t, a_t, \mathcal{D}] = \mathbb{E}_{p(\theta \mid \mathcal{D})}[\mathrm{Var}[s_{t+1} \mid s_t, a_t; \theta]] + \mathrm{Var}_{p(\theta \mid \mathcal{D})}[\mathbb{E}[s_{t+1} \mid s_t, a_t; \theta]]$$

$$= \underbrace{\mathbb{E}_\theta[\sigma^2_{ale}(s_t, a_t; \theta)]}_{\text{incertezza aleatoria}} + \underbrace{\sigma^2_{epi}(s_t, a_t)}_{\text{incertezza epistemica}}$$

Il termine epistemico σ^2_{epi} è la varianza delle predizioni medie tra diversi parametri θ; il termine aleatoria σ^2_{ale} è la varianza media delle previsioni di ciascun modello. Un ensemble di N modelli stima σ^2_{epi} come la varianza delle predizioni tra i modelli, e σ^2_{ale} come la media delle varianze individuali.

La distinzione tra EU e AU è operativamente importante per l'esplorazione in RL model-based: l'agente dovrebbe eseguire azioni che riducono l'EU (esplorazione epistemica) nei stati dove la dinamica è mal compresa, e l'AU informa il pianificatore sull'incertezza intrinseca che deve essere considerata nel piano. Politiche di esplorazione basate su EU, come PETS (Probabilistic Ensembles with Trajectory Sampling, Chua et al. 2018), mostrano una notevole efficienza di campionamento rispetto ai metodi deterministici.

4. Pianificazione nello spazio delle belief con modelli neurali

L'integrazione tra POMDP e modelli del mondo neurali è realizzata nell'architettura MERLIN (Memory, RL and Inference Network, Wayne et al. 2018) e nelle sue varianti. MERLIN mantiene uno stato di memoria che gioca il ruolo della belief state, aggiornandola con un meccanismo di scrittura differenziabile ispirato alle reti di memoria esterna.

La belief state neurale è un vettore $h_t \in \mathbb{R}^d$ aggiornato da un *belief update network* f_{BU}:

$$h_t = f_{BU}(h_{t-1}, z_t, a_{t-1})$$

dove z_t è la rappresentazione latente dell'osservazione corrente (dall'encoder). L'aggiornamento non è esplicitamente bayesiano, ma si addestra end-to-end a produrre stati interni sufficienti per la pianificazione ottimale. Questo approccio rinuncia alla garanzia formale di ottimalità del belief update bayesiano in favore di un meccanismo flessibile apprezzato dalla rete come più informativo per la politica.

Una variante più fedele al framework POMDP è offerta dalle *Deep POMDP* (Igl et al. 2018), che mantengono una distribuzione esplicita sullo stato latente z_t attraverso un filtro particellare differenziabile. Il filtro particellare neurale rappresenta la belief come un insieme di K particelle $\{z_t^k, w_t^k\}_{k=1}^K$ con pesi associati, e aggiorna le particelle seguendo lo schema standard del Sequential Importance Resampling (SIR):

Predizione: $z_{t+1}^k \sim p_\theta(z \mid z_t^k, a_t)$ — passo di predizione
Pesatura: $\tilde{w}_{t+1}^k = w_t^k \cdot p_\varphi(o_{t+1} \mid z_{t+1}^k)$ — likelihood ponderata
Normalizzazione: $w_{t+1}^k = \tilde{w}_{t+1}^k / \Sigma_j \tilde{w}_{t+1}^j$ — normalizzazione
Ricampionamento: resample se $ESS = (\Sigma_k w_k)^2 / \Sigma_k w_k^2 < K/2$ — degeneracy check

Il gradiente del loss rispetto ai parametri del modello di transizione e di osservazione fluisce attraverso le particelle tramite la tecnica del *soft resampling* (Karkus et al. 2018): invece di un'operazione di selezione discreta, il ricampionamento viene sostituito da un ricampionamento soft differenziabile che combina le particelle con pesi morbidi:

$$z_{t+1}^k \leftarrow \Sigma_j \, \mathrm{softmax}(\alpha \, w_j) \cdot z_{t+1}^j$$

dove α è un parametro di temperatura che controlla il trade-off tra ricampionamento hard ($\alpha \to \infty$) e soft ($\alpha \to 0$).

5. Inferenza attiva: il principio di energia libera come framework unificante

Il principio dell'energia libera (*Free Energy Principle*, FEP) di Karl Friston, introdotto nella Parte I in forma sommaria, merita un'analisi matematica molto più approfondita, in quanto costituisce forse il framework teorico più ambizioso per unificare percezione, apprendimento e azione in un unico principio variazionale.

Il FEP postula che ogni sistema biologico (e per estensione artificiale) è organizzato in modo da minimizzare la sua energia libera variazionale F:

$$F[q, o] = \mathbb{E}_{q(s)}[\log q(s) - \log p(o, s)]$$
$$= KL[q(s) \,||\, p(s \mid o)] - \log p(o)$$

dove $q(s)$ è la distribuzione approssimante (credenza interna), $p(o, s) = p(o \mid s)\, p(s)$ è il modello generativo, e $p(o)$ è la evidenza (sorpresa logaritmica negativa sull'osservazione o). Il FEP afferma che un agente minimizza F sia rispetto a q (inferenza, percezione) sia rispetto alle azioni che modificano o (azione, comportamento).

Il collegamento con il framework POMDP è profondo: la minimizzazione di F rispetto a q equivale al belief update bayesiano (quando Q è ottimale, KL[q || p(s|o)] = 0 e F = −log p(o)). La minimizzazione di F rispetto alle azioni equivale alla selezione di azioni che minimizzano la sorpresa attesa sulle osservazioni future — il che corrisponde, in alcuni regimi, alla massimizzazione dell'utilità attesa e, in altri, alla minimizzazione dell'entropia delle osservazioni future.

L'inferenza attiva (*Active Inference*, AI) è la variante del FEP che incorpora la previsione del futuro. L'agente non solo minimizza l'energia libera rispetto alle osservazioni correnti, ma anche rispetto alle osservazioni future attese sotto diversi piani d'azione. Per fare ciò, introduce una distribuzione sulle traiettorie future $\tilde{q}(\tilde{a}, \tilde{o})$ e una distribuzione a priori $\tilde{Q}(\tilde{a}, \tilde{o})$ che codifica le preferenze dell'agente (gli stati obiettivo):

$$G(\pi) = \mathbb{E}_{\tilde{q}(o, s \mid \pi)}[\log \tilde{q}(s \mid \pi) - \log \tilde{p}(o, s)]$$

dove $G(\pi)$ è la *expected free energy* (EFE) della politica π, e $\tilde{p}(o, s)$ include sia il modello generativo sia la distribuzione di preferenze sulle osservazioni. Si può dimostrare che la EFE si scompone in due termini:

$$\begin{aligned} G(\pi) = &- \mathbb{E}_{\tilde{q}(o|\pi)}[\log \tilde{p}(o)] && \text{— valore pragmatico (utilità attesa)} \\ &- \mathbb{E}_{\tilde{q}(s|\pi)}[H[p(o \mid s)]] && \text{— valore epistemico (curiosità)} \\ &+ KL[\tilde{q}(s \mid o, \pi) \,||\, \tilde{q}(s \mid \pi)] && \text{— informazione attesa} \end{aligned}$$

Il primo termine misura quanto le osservazioni future attese concordano con le preferenze (stati obiettivo); il secondo misura la riduzione attesa dell'incertezza sul mondo (informazione guadagnata dalle azioni); il terzo è un termine di risoluzione dell'incertezza (*ambiguity*). Le politiche che minimizzano G bilanciano pertanto il raggiungimento degli obiettivi (utilità) con l'esplorazione informativa (curiosità) — una proprietà che emerge naturalmente dall'ottimizzazione della EFE senza necessità di bonus di esplorazione aggiuntivi.

6. Connessione tra Active Inference e apprendimento per rinforzo

La relazione tra AI e RL profondo è stata analizzata rigorosamente da Millidge (2020) e da Sajid e collaboratori (2021). Si dimostra che l'RL standard (massimizzazione del ritorno atteso scontato) è un caso limite dell'AI quando la distribuzione di preferenze $\tilde{p}(o)$ è una delta di Dirac sugli stati obiettivo e il termine epistemico è trascurato.

La differenza fondamentale sta nella *costruzione esogena vs. endogena degli obiettivi*: l'RL standard riceve una funzione di ricompensa estrinseca R(s, a) definita dall'utente, mentre l'AI deriva gli obiettivi comportamentali endogenamente dalla struttura del modello generativo e delle preferenze a priori. In sistemi robotici pratici, questa distinzione è rilevante: l'AI permette di specificare obiettivi come distribuzioni sulle osservazioni desiderate (ad esempio, "l'oggetto dovrebbe trovarsi sulla mensola")

piuttosto che come funzioni di ricompensa numeriche, che richiedono una ingegnerizzazione attenta (*reward engineering*).

Dal punto di vista computazionale, l'inferenza attiva richiede la valutazione della EFE per ogni politica candidata, il che è intrattabile per spazi di azioni continui e orizzonti lunghi. I *Deep Active Inference* models (Çatal et al. 2020, Fountas et al. 2020) approssimano la valutazione della EFE attraverso reti neurali addestrate su traiettorie campionate, avvicinando la complessità computazionale a quella del RL profondo model-based.

7. Il problema dell'esplorazione efficiente: curiosità, novità e information gain

Il problema dell'esplorazione — come l'agente dovrebbe bilanciare lo sfruttamento della conoscenza acquisita con l'acquisizione di nuova informazione — è uno dei problemi centrali del RL ed è strettamente connesso alla teoria dei modelli del mondo. Un agente dotato di un modello del mondo accurato può valutare l'incertezza sulle transizioni e guidare l'esplorazione verso le regioni di stato dove l'incertezza è più alta (esplorazione diretta dall'incertezza, *uncertainty-driven exploration*).

Il bonus di curiosità (*curiosity bonus*) più influente è il *Random Network Distillation* (RND) di Burda e collaboratori (2019), che misura la novità di uno stato come l'errore di predizione di una rete neurale addestrata a imitare le uscite di una rete fissata e casuale:

$$r_{curiosity}(s) = ||f_\theta(s) - f_{fixed}(s)||_2^2$$

dove f_{fixed} è una rete con pesi fissati casualmente e f_θ è addestrata on-line a predire le uscite di f_{fixed}. Poiché f_θ viene addestrata solo sugli stati visitati, l'errore di predizione è alto per gli stati nuovi e basso per gli stati frequentemente visitati. RND è computazionalmente leggero e non richiede un modello del mondo esplicito.

Il *Prediction Gain* (Still e Precup, 2012) e il *Bayesian Surprise* (Baldi e Itti, 2010) formalizzano la curiosità come l'informazione guadagnata da una nuova osservazione rispetto al modello corrente:

$$r_{surprise}(o_t) = KL[p(\theta \mid o_{1:t}) \,||\, p(\theta \mid o_{1:t-1})]$$

che misura quanto la nuova osservazione cambia la credenza sui parametri del modello. Questa quantità è il *Bayesian information gain* e corrisponde esattamente al termine epistemico nella EFE dell'inferenza attiva.

VIII. Leggi di scala, capacità emergenti e modelli fondazionali del mondo

1. Le leggi di scala neurale: fondamenti teorici

La scoperta che le prestazioni dei modelli neurali crescono in modo prevedibile e regolare con le risorse computazionali e la quantità di dati — le cosiddette *scaling laws* — ha trasformato radicalmente la pratica dell'apprendimento automatico nell'ultimo decennio. Kaplan e collaboratori (2020, OpenAI) stabilirono empiricamente che la loss di test L di un modello linguistico segue una legge di potenza in tre fattori indipendenti: numero di parametri N, dimensione del dataset D, e budget computazionale di training C:

$$L(N) \sim (N_c / N)^{\alpha_N}$$
$$L(D) \sim (D_c / D)^{\alpha_D}$$
$$L(C) \sim (C_c / C)^{\alpha_C}$$

con esponenti empiricamente misurati $\alpha_N \approx 0.076$, $\alpha_D \approx 0.095$, $\alpha_C \approx 0.050$ per modelli linguistici. Le costanti N_c, D_c, C_c dipendono dall'architettura e dal dominio. Crucialmente, queste leggi indicano che il ritorno marginale diminuisce (*diminishing returns*) ma non si azzera: raddoppiare il numero di parametri riduce sempre la loss di un fattore costante.

Hoffmann e collaboratori (2022, DeepMind) hanno poi raffinato queste leggi con il criterio di *compute-optimal training* (modello Chinchilla): dato un budget computazionale fisso C, la configurazione ottimale (N*, D*) bilancia parametri e dati nel rapporto $D^* \approx 20\, N^*$, violando la pratica allora comune di addestrare modelli molto grandi su dataset relativamente piccoli. La derivazione segue da:

$$C = 6ND \quad \text{(costo approssimato del training: 6 FLOPs per parametro per token)}$$
$$L(N, D) = E + A/N^{\alpha_N} + B/D^{\alpha_D}$$

Minimizzando L soggetto al vincolo C = 6ND con i moltiplicatori di Lagrange:

$$\partial L/\partial N = -\alpha_N A / N^{\alpha_N + 1} + \lambda \cdot 6D = 0$$
$$\partial L/\partial D = -\alpha_D B / D^{\alpha_D + 1} + \lambda \cdot 6N = 0$$

Da cui: $N^{\alpha_N + 1} / D^{\alpha_D + 1} = \alpha_N A / (\alpha_D B) =$ costante, che, combinato con C = 6ND, dà la relazione ottimale $N^* \propto C^{\alpha_D/(\alpha_N + \alpha_D)}$ e $D^* \propto C^{\alpha_N/(\alpha_N + \alpha_D)}$.

Per i modelli del mondo visivi, le leggi di scala sono state studiate da Zhai e collaboratori (2022) per i Vision Transformer e da Henighan e collaboratori (2020) per i modelli autoregressivi su immagini. Gli esponenti risultano diversi da quelli linguistici, riflettendo le differenti strutture statistiche dell'informazione visiva rispetto a quella testuale.

2. Capacità emergenti: dalla quantità alla qualità

Le leggi di scala prevedono una riduzione continua e monotona della loss, ma le prestazioni su task specifici mostrano talvolta un comportamento radicalmente non lineare: una *capacità emergente* è una competenza che il modello non possiede affatto (prestazione al livello del caso) fino a una certa soglia di scala, oltre la quale compare bruscamente. Wei e collaboratori (2022) documentano decine di tali capacità nei grandi modelli linguistici: ragionamento aritmetico a catena, traduzione in lingue a bassa risorsa, risoluzione di analogie complesse.

La spiegazione teorica di queste emergenze è ancora dibattuta. Schaeffer, Miranda e Koyejo (2023) argomentano convincentemente che molte emergenze apparenti sono artefatti della metrica di valutazione: con metriche a soglia (come l'accuratezza binaria), una miglioramento continuo sottostante della probabilità corretta produce un'apparente discontinuità nell'accuratezza quando questa attraversa la soglia del 50%. Con metriche continue come la log-probabilità o la BLEU score, la curva di scala rimane monotona. Tuttavia, alcune emergenze sembrano resistere a questa critica — in particolare quelle legate a ricombinazione compositiva di conoscenze distanti.

Per i modelli del mondo, la questione delle capacità emergenti è particolarmente rilevante: modelli addestrati su grandi quantità di dati video (come Genie, SoRA, UniSim) mostrano la comparsa, oltre certe soglie di scala, di capacità di generalizzazione fisica che sembrano qualitative piuttosto che quantitative. La comprensione della causalità fisica, la resistenza all'occlusione, la previsione delle conseguenze di azioni meccaniche — queste competenze emergono in modo discontinuo al crescere della scala, suggerendo che raggiungono una rappresentazione interna qualitativamente diversa dalla semplice interpolazione statistica.

3. Genie: generazione interattiva di ambienti

Genie (Generative Interactive Environments, Bruce et al. 2024, Google DeepMind) è un modello fondazionale del mondo addestrato su grandi quantità di video di videogiochi 2D senza alcuna annotazione di azioni. L'architettura produce un ambiente interattivo: dato un frame iniziale e una sequenza di comandi latenti, il modello genera coerentemente i frame successivi, simulando un ambiente con fisica plausibile.

La struttura di Genie si compone di tre moduli. Il primo è un tokenizer spazio-temporale (un VQVAE 3D) che comprime sequenze di frame in sequenze di token discreti. Il secondo è il modello di azione latente: a partire da coppie di frame consecutivi, il modello inferisce l'azione latente $a_t \in \{1, ..., K\}$ (variabile discreta) che "spiega" la transizione — senza che questa azione sia osservata direttamente. Il terzo è un modello dinamico (un Transformer autoregressivo) che predice il token del frame successivo dato il token del frame corrente e l'azione latente.

Il modello di azione latente è addestrato con la tecnica del *Video Action Coding*, che è una variante dell'apprendimento contrastivo temporale: l'azione latente deve essere predittiva della transizione da un frame all'altro, ma non del contenuto assoluto dei frame. Formalmente:

$$a_t = \text{argmax}_{k \in \{1,...,K\}} \exp(E_a(\text{frame}_t, \text{frame}_{t+1}) \cdot e_k) / \Sigma_j \exp(E_a(\text{frame}_t, \text{frame}_{t+1}) \cdot e_j)$$

dove E_a è un encoder di transizione e $\{e_k\}$ sono i vettori di embedding delle azioni latenti. Il modello è addestrato con InfoNCE usando come negativi frame di transizione diverse.

Il risultato sorprendente è che le azioni latenti apprese corrispondono in modo interpretabile a comandi di gioco naturali (muoversi a sinistra/destra, saltare, interagire) senza alcun supervisione — un esempio di *disentanglement* emergente tra la struttura dell'azione e quella dello stato.

4. UniSim: simulatori universali come modelli del mondo

UniSim (Universal Simulator, Yang et al. 2023) adotta la prospettiva che un simulatore del mondo fisico sufficientemente accurato e generalista potrebbe essere usato come ambiente di training per qualsiasi politica robotica, eliminando la necessità di simulatori fisici specializzati e costosi.

L'architettura UniSim è un video diffusion model condizionato su testo e azioni, che sintetizza video coerenti di scene fisiche da un prompt testuale e da una sequenza di azioni robotiche. La condizionalità sull'azione è implementata attraverso cross-attention tra le feature del video e gli embedding delle azioni, permettendo al modello di generare video che riflettono realisticamente le conseguenze delle azioni nel mondo fisico.

La loss di addestramento segue quella dei modelli di diffusione con condizionamento:

$$\mathcal{L}_{\text{UniSim}} = \mathbb{E}_{t, x_0, \varepsilon}[||\varepsilon - \varepsilon_\theta(x_t, t, c_{\text{text}}, c_{\text{action}})||_2^2]$$

dove c_{text} e c_{action} sono gli embedding del testo e dell'azione, $x_t = \sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{(1-\bar{\alpha}_t)}\, \varepsilon$ è il frame con rumore, e ε_θ è la rete di denoising (tipicamente un U-Net con attenzione). Durante l'addestramento, le azioni vengono droppate con probabilità p_{drop} per permettere il guidance senza classificatore (*classifier-free guidance*) che migliora la qualità della generazione a costo di ridotta diversità.

5. SORA e i video diffusion model come world model impliciti

SORA (OpenAI, 2024) è il più visibile rappresentante dei video diffusion model su larga scala, capace di generare video di alta qualità di durata fino a 60 secondi con coerenza temporale notevole. Sebbene OpenAI non abbia pubblicato i dettagli architetturali completi, la nota tecnica disponibile descrive l'uso di patch spazio-temporali come token per un Transformer di diffusione (*Diffusion Transformer*, DiT, Peebles e Xie 2023).

Il DiT sostituisce la backbone U-Net dei modelli di diffusione classici con un Transformer che opera su una sequenza di patch latenti del video. La chiave è che le patch spazio-temporali hanno dimensione fissa indipendentemente dalla risoluzione e dalla durata del video, rendendo il modello *natively flexible* rispetto a questi parametri: addestrando su video di risoluzioni e durate diverse, il modello apprende a gestire sequenze di token di lunghezza variabile.

Dal punto di vista dei modelli del mondo, SORA e sistemi analoghi (Emu Video, Lumiere, Wan) costituiscono world model impliciti: non sono progettati esplicitamente per la pianificazione o il controllo, ma le rappresentazioni interne che sviluppano per produrre video fisicamente coerenti codificano necessariamente strutture dinamiche sul mondo. Il fatto che SORA possa simulare correttamente la riflessione di oggetti sulla superficie dell'acqua, la deformazione di materiali morbidi, o le interazioni inerziali tra corpi suggerisce l'emergenza di una comprensione implicita della fisica di Newton — anche se questa conoscenza è distribuita e non simbolicamente accessibile.

6. Scaling dei modelli del mondo per la robotica: RT-X e Open X-Embodiment

La sfida principale dei modelli fondazionali per la robotica è la scarsità dei dati: mentre i modelli linguistici e visivi possono sfruttare trilioni di token di testo e miliardi di immagini da internet, i dati di interazione robotica (traiettorie sensorimotorie con reward o dimostrazioni) sono ordini di grandezza più scarsi e costosi da raccogliere.

Il progetto *Open X-Embodiment* (Collaboration et al. 2023) ha aggregato dataset robotici da 22 laboratori diversi, con robot di morfologie diverse (braccia, mobile manipulators, robot bimanuali), task diversi e configurazioni di sensori diverse, per un totale di oltre un milione di traiettorie reali. RT-X è il modello addestrato su questo dataset, e mostra una generalizzazione notevolmente superiore a modelli addestrati su singoli dataset, confermando che la diversità dei dati è cruciale per la generalizzazione robotica.

La questione tecnica chiave è come gestire l'eterogeneità morfologica: robot con diverso numero di gradi di libertà, diversi spazi di azione (joint space, end-effector space, delta-position), diversi sensori (mono/stereo camera, force sensor, tactile sensor). RT-X usa una rappresentazione delle azioni tokenizzata e discretizzata che

unifica spazi di azione eterogenei: ogni dimensione dell'azione è binned in 256 bin e codificata come token, rendendo il problema di control equivalente al problema di language modeling — predizione autoregressiva del token successivo.

7. Il teorema di approssimazione universale per le sequenze e limiti profondi

Il teorema di approssimazione universale (*Universal Approximation Theorem*, UAT) per le reti neurali feedforward (Cybenko 1989, Hornik 1991) afferma che una rete con un singolo strato nascosto sufficientemente largo può approssimare uniformemente qualsiasi funzione continua su un compatto di ℝ^n. Per i Transformer, Yun e collaboratori (2020) hanno dimostrato un UAT per sequenze: un Transformer a due strati con attenzione multi-testa e FFN può approssimare uniformemente qualsiasi funzione continua su sequenze di lunghezza fissa.

Tuttavia, questi risultati di approssimabilità non implicano apprendibilità: la funzione ottimale può esistere nella classe del modello, ma l'ottimizzazione con discesa del gradiente stocastica può non trovarla in tempo polinomiale. I risultati teorici sull'ottimizzazione dei Transformer (Tian et al. 2023, Mahankali et al. 2023) mostrano che in certi regimi — in-context learning su distribuzioni semplici, regressione lineare — la discesa del gradiente converge alla soluzione ottimale. Per i problemi di modellazione del mondo con struttura arbitraria, la convergenza globale rimane aperta.

Il limite fondamentale dei modelli del mondo basati su Transformer per il ragionamento fisico riguarda la composizionalità spaziale e causale: un Transformer addestrato su video di oggetti che cadono impara la distribuzione statistica delle traiettorie di caduta, ma non necessariamente il principio fisico (F = ma) che le genera. La generalizzazione a configurazioni radicalmente nuove (oggetti con densità anomala, gravità alternativa) richiede una forma di astrazione causale che i modelli attuali possono o non possono avere sviluppato. Questa distinzione tra *correlazione statistica* e *comprensione causale* rimane il problema aperto più profondo nella ricerca sui modelli del mondo.

8. Latent World Model e il problema della grounding fisico

Il problema del *grounding fisico* — la connessione tra i simboli interni di un modello e le proprietà fisiche del mondo reale — è particolarmente acuto per i modelli del mondo latenti. Una rappresentazione z_t che predice accuratamente z_{t+1} per traiettorie viste durante l'addestramento può farlo attraverso correlazioni superficiali (*texture shortcuts*, *color cues*, *background statistics*) piuttosto che attraverso la comprensione delle strutture causali sottostanti.

Il *systematic generalization* testing, proposto da Lake e Baroni (2018) per il linguaggio e esteso alla visione da Bahdanau e collaboratori (2020), valuta la capacità del modello di generalizzare correttamente a composizioni di concetti visti separatamente ma mai in combinazione durante l'addestramento. Per i modelli del mondo fisici, questo si traduce in test come: "dato un modello addestrato su sfere che rotolano e cubi che scivolano, generalizza correttamente a sfere che scivolano e cubi che rotolano?". I risultati attuali mostrano che i modelli neurali più capaci falliscono sistematicamente in questi test, confermando che il grounding fisico rimane un problema aperto.

Le architetture strutturate — come i modelli a slot (Slot Attention, SlotFormer) e i modelli a grafo — mostrano miglioramenti significativi nella generalizzazione compositiva rispetto ai modelli monolitici, ma il gap rispetto alla generalizzazione umana rimane ampio. La tesi di LeCun, secondo cui la predizione nello spazio latente (JEPA) è necessaria ma forse non sufficiente per il grounding fisico, e che sono necessarie rappresentazioni strutturate gerarchiche, sembra la più coerente con i dati empirici disponibili, anche se non è ancora stata dimostrata in modo convincente.

IX. Geometria differenziale riemanniana e information geometry per l'apprendimento delle rappresentazioni

1. Varietà differenziabili e strutture metriche nello spazio latente

La tradizione dominante dell'apprendimento automatico tratta gli spazi latenti come sottospazi di $\mathbb{R}^n$ dotati della metrica euclidea standard, una convenzione che, pur notevolmente produttiva, introduce distorsioni sistematiche nella geometria delle rappresentazioni apprese. Le distanze euclidee misurano la discrepanza tra coordinate, ma le coordinate di uno spazio latente appreso non hanno un significato metrico intrinseco: due punti vicini in norma L_2 non sono necessariamente semanticamente prossimi, e viceversa. La geometria differenziale riemanniana offre un apparato concettuale e computazionale per trattare gli spazi latenti come oggetti geometrici intrinsecamente curvi, in cui le nozioni di distanza, interpolazione e gradiente vengono ridefinite in modo più fedele alla struttura semantica sottostante.

Una varietà differenziabile $\mathcal{M}$ di dimensione d è uno spazio topologico di Hausdorff con un atlante di carte locali $\{(U\alpha, \varphi\alpha)\}$, dove ciascuna carta $\varphi\alpha : U\alpha \to \mathbb{R}^d$ è un omeomorfismo e le transizioni $\varphi\beta \circ \varphi\alpha^{-1}$ tra carte sovrapposte sono diffeomorfismi di classe C^∞. Questa struttura permette di esportare su $\mathcal{M}$ il calcolo differenziale di $\mathbb{R}^d$ in modo invariante rispetto alla scelta delle coordinate locali. In ogni punto $p \in \mathcal{M}$ si definisce lo spazio tangente T M come lo spazio vettoriale delle derivazioni sull'algebra delle funzioni lisce in p, canonicamente isomorfo a $\mathbb{R}^d$ tramite qualsiasi carta locale.

Una varietà riemanniana è una coppia $(\mathcal{M}, g)$ dove g è un tensore metrico: un'assegnazione liscia $p \mapsto g$ di forme bilineari simmetriche definite positive su $T\mathcal{M}$. Il tensore metrico g fornisce un prodotto interno sullo spazio tangente in ogni punto, da cui discendono nozioni di lunghezza dei vettori tangenti, angoli tra curve, e lunghezza dei cammini. La lunghezza di un cammino liscio $\gamma : [0,1] \to \mathcal{M}$ è:

$$L(\gamma) = \int_0^1 \sqrt{(g_{\gamma(t)}(\gamma'(t), \gamma'(t)))}\, dt$$

La distanza geodetica $d_g(p, q)$ tra due punti è l'infimo delle lunghezze di tutti i cammini lisci che li connettono:

$$d_g(p, q) = \inf \{ L(\gamma) : \gamma(0) = p, \gamma(1) = q \}$$

Le geodetiche --- le curve che realizzano localmente questa distanza minima --- soddisfano l'equazione delle geodetiche in coordinate locali:

$$\frac{d^2 x^k}{dt^2} + \Gamma^k_{ij} \frac{dx^i}{dt}\frac{dx^j}{dt} = 0$$

dove i Γ^k_{ij} sono i simboli di Christoffel di Levi-Civita, costruiti a partire dalle derivate del tensore metrico:

$$\Gamma^k_{ij} = \tfrac{1}{2} g^{kl} \left(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij} \right)$$

e dove si usa la convenzione di Einstein per la somma sugli indici ripetuti. I simboli di Christoffel codificano la curvatura intrinseca della varietà: su $\mathbb{R}^n$ euclideo tutti i Γ si annullano e l'equazione geodetica diventa $d^2x/dt^2 = 0$, ossia le rette rettilinee.

2. La metrica di Fisher come tensore riemanniano sulle famiglie statistiche

Il collegamento più profondo tra geometria riemanniana e apprendimento automatico transita attraverso l'information geometry, disciplina inaugurata da Shun'ichi Amari negli anni Ottanta e sistematizzata nella monografia del 2000 insieme a Nagaoka. L'intuizione centrale è che uno spazio di distribuzioni di probabilità parametriche $\{p(x;\theta) : \theta \in \Theta\}$ può essere dotato naturalmente di una struttura riemanniana, la cui metrica è la matrice di informazione di Fisher.

Si consideri una famiglia liscia di distribuzioni $p(x;\theta)$ su uno spazio campionario $\mathcal{X}$, con $\theta \in \Theta \subseteq \mathbb{R}^m$. La matrice di informazione di Fisher $F(\theta)$ è definita come:

$$F_{ij}(\theta) = \mathbb{E}_{x \sim p(\cdot;\theta)}\left[\partial_i \log p(x;\theta) \cdot \partial_j \log p(x;\theta) \right]$$

$$= -\,\mathbb{E}_{x \sim p(\cdot;\theta)}\left[\partial_i \partial_j \log p(x;\theta) \right]$$

dove $\partial_i = \partial/\partial\theta_i$. Questa matrice è simmetrica e semidefinita positiva (definita positiva sotto condizioni di regolarità), e dunque definisce un tensore metrico riemanniano su Θ. La metrica di Fisher $g_{ij}(\theta) = F_{ij}(\theta)$ è l'unica metrica riemanniana sullo spazio delle distribuzioni che sia invariante rispetto a reparametrizzazioni sufficienti e rispetto

all'azione dei morfismi stocastici --- un risultato noto come teorema di Chentsov (1972). Questa unicità conferisce alla metrica di Fisher un carattere intrinseco che le metriche euclidee sullo spazio dei parametri non possono avere.

La distanza geodetica indotta dalla metrica di Fisher non coincide con nessuna divergenza standard, ma è strettamente correlata alla divergenza di Kullback-Leibler: per distribuzioni vicine, la divergenza KL si approssima come:

$$KL[p(x;\theta) \,\|\, p(x;\theta + d\theta)] = (1/2)\, F_i\square(\theta)\, d\theta^{\mathbf{i}}\, d\theta^{j} + O(\|d\theta\|^{3})$$

La divergenza KL è pertanto il quadrato della distanza di Fisher al secondo ordine infinitesimale. Questa relazione spiega perché la metrica di Fisher è lo strumento naturale per l'ottimizzazione in spazi di distribuzioni: il gradiente naturale, che corregge il gradiente euclideo attraverso la metrica di Fisher, minimizza la loss nella metrica statisticamente corretta.

3. Il gradiente naturale e la sua connessione con i modelli del mondo

Il gradiente naturale, introdotto da Amari (1998), è il gradiente di una funzione scalare $F(\theta)$ rispetto alla metrica di Fisher:

$$\nabla\tilde{}F(\theta) = F(\theta)^{-1}\, \nabla F(\theta)$$

dove $\nabla F(\theta)$ è il gradiente euclideo ordinario e $F(\theta)^{-1}$ è l'inversa della matrice di Fisher. La discesa del gradiente naturale aggiorna i parametri secondo:

$$\theta_\{t+1\} = \theta \; - \eta\, F(\theta)^{-1}\, \nabla F(\theta)$$

L'interpretazione geometrica è immediata: invece di muoversi di un passo proporzionale al gradiente euclideo nello spazio dei parametri, il gradiente naturale si muove di un passo proporzionale al gradiente riemanniano nello spazio statistico, ossia nella direzione di massima pendenza rispetto alla distanza di Fisher. Questo lo rende invariante rispetto a reparametrizzazioni del modello: se si sostituisce θ con un'altra parametrizzazione $\eta = \varphi(\theta)$, il gradiente euclideo cambia, ma il gradiente naturale non cambia, perché la metrica di Fisher si trasforma contravariante in modo da compensare.

Per i modelli a variabile latente impiegati nei world model, il gradiente naturale presenta vantaggi teorici sostanziali. In un modello come l'RSSM, la funzione di perdita dipende dalla distribuzione posteriore $q_\varphi(z_t \mid h_t, o_t)$ e dal prior $p_\varphi(z_t \mid h_t)$. Il gradiente euclideo rispetto a φ può risultare mal condizionato se le parametrizzazioni delle medie e delle varianze delle gaussiane hanno scale molto diverse: piccole variazioni nei parametri della covarianza producono grandi cambiamenti nella distribuzione, mentre variazioni della stessa grandezza nei parametri della media producono cambiamenti minori. Il gradiente naturale normalizza automaticamente queste scale attraverso la geometria della famiglia di distribuzioni.

Il principale ostacolo computazionale è il calcolo e l'inversione della matrice di Fisher $F(\theta) \in \mathbb{R}^{n\times n}$, dove n può essere nell'ordine dei miliardi per reti profonde moderne. Le approssimazioni pratiche includono la fattorizzazione di Kronecker KFAC (Martens e Grosse, 2015), che approssima la Fisher di ogni strato come prodotto Kronecker di due matrici più piccole:

$$F \approx \otimes_{\ell} (A^{l} \otimes G^{l})$$

dove A^{l} è la covarianza degli input al livello ℓ e G^{l} è la covarianza dei gradienti dello stesso livello. KFAC rende il gradiente naturale computazionalmente trattabile per reti di dimensioni moderate, con un overhead di memoria e calcolo di circa un fattore 3–5 rispetto alla discesa del gradiente ordinaria.

4. Varietà riemanniane apprese: pullback metrics e geometria indotta dalle reti neurali

Una prospettiva complementare considera non una varietà fissa dotata di una metrica, bensì la geometria che una rete neurale induce implicitamente sullo spazio degli input attraverso la sua funzione di codifica. Si consideri un encoder $f_\theta : \mathcal{X} \to \mathbb{R}^{d}$ che mappa osservazioni in un spazio latente euclideo. La metrica di pullback di f_θ è il tensore metrico su $\mathcal{X}$ definito da:

$$g^{pullback}_x(v, w) = \langle J_\theta(x)v, J_\theta(x)w \rangle = v^T J_\theta(x)^T J_\theta(x)\, w$$

dove $J_\theta(x) = \partial f_\theta / \partial x \in \mathbb{R}^{d\times \dim(\mathcal{X})}$ è la matrice jacobiana di f_θ in x, e $\langle \cdot,\cdot \rangle$ è il prodotto interno euclideo nello spazio latente. Questa metrica cattura esattamente le distorsioni che l'encoder introduce: direzioni nello spazio degli input lungo le quali l'encoder varia rapidamente hanno grande norma rispetto a $g^{pullback}$, mentre direzioni invisibili all'encoder (il suo nucleo) hanno norma nulla.

La metrica di pullback ha una rilevanza diretta per l'analisi delle rappresentazioni apprese da JEPA e da metodi contrastivi. Se l'encoder f_θ della JEPA apprende a ignorare i dettagli irrilevanti per la predizione latente, le direzioni corrispondenti a queste variazioni irrilevanti appartengono al nucleo di J_θ e hanno norma nulla rispetto a $g^{pullback}$. La varietà appresa è quindi una proiezione dello spazio degli input sulle direzioni predittivamente rilevanti, formalizzando in termini geometrici precisi l'intuizione di LeCun sulla necessità di scartare i dettagli impredicibili.

Hauberg e collaboratori (2012) hanno proposto di usare le metriche di pullback per definire geodetiche nello spazio degli input che siano semanticamente significative: invece di interpolare linearmente tra due immagini (il che produce artefatti visivi non realistici), si calcola la geodetica rispetto alla metrica di pullback dell'encoder, ottenendo una traiettoria che rimane sulla varietà dei dati semanticamente coerenti. L'equazione geodetica per la metrica di pullback diventa in coordinate dell'input:

$$d^2x^k/dt^2 + (J^T J)^{kl} \left[(\partial_i (J^T J)_{lj} - \tfrac{1}{2}\, \partial_l (J^T J)_{ij})\, (dx^i/dt)(dx^j/dt)\right] = 0$$

La soluzione numerica di questa equazione richiede il calcolo delle derivate seconde dell'encoder, che in reti profonde si ottiene attraverso differenziazione automatica al secondo ordine (ad esempio con lo Hessian-vector product in PyTorch o JAX).

5. Geometria iperbolica per le rappresentazioni gerarchiche

Un caso particolarmente importante di geometria non euclidea per l'apprendimento delle rappresentazioni è lo spazio iperbolico $\mathbb{H}^n$, la varietà riemanniana a curvatura sezionale costante negativa $-1/c$ (con $c > 0$). Lo spazio iperbolico presenta proprietà di espansione volumetrica esponenziale con la distanza dal centro --- analoga alla crescita esponenziale di un albero con la profondità --- che lo rendono naturalmente adatto a codificare gerarchie e strutture arborescenti.

Il modello di Poincaré del disco è la realizzazione di $\mathbb{H}^2$ come il disco unitario aperto $\mathbb{D} = \{x \in \mathbb{R}^2 : \|x\| < 1\}$ dotato della metrica conforme:

$$g^{\mathbb{H}}_x = (2c / (1 - c\|x\|^2))^2 \, g^{eucl}_x$$

dove g^{eucl} è la metrica euclidea standard. Il fattore di conformalità $(2c/(1 - c\|x\|^2))^2$ diverge al bordo del disco ($\|x\| \to 1$), il che significa che le distanze iperboliche crescono molto più rapidamente delle distanze euclidee verso la periferia. La distanza geodetica iperbolica tra due punti $x, y \in \mathbb{D}$ è:

$$d_{\mathbb{H}}(x, y) = (1/\sqrt{c}) \operatorname{arccosh}(1 + 2c \, \|x - y\|^2 / ((1 - c\|x\|^2)(1 - c\|y\|^2)))$$

Nickel e Kiela (2017) hanno mostrato che embeddings in $\mathbb{H}^n$ di dimensione anche molto ridotta ($n = 2$ o $n = 5$) possono codificare gerarchie lessicali con distorsione drasticamente inferiore rispetto agli embedding euclidei di dimensione analoga. L'intuizione geometrica è che il centro del disco rappresenta la radice della gerarchia, e i nodi figli vengono collocati più vicini alla periferia, sfruttando l'espansione esponenziale dello spazio per separare i rami della gerarchia senza interferenza reciproca.

Per i modelli del mondo con struttura gerarchica --- come il modello multi-scala di LeCun in cui diversi livelli della gerarchia JEPA operano a scale temporali diverse --- lo spazio iperbolico offre un'arena geometrica naturale. La JEPA di basso livello opera nelle vicinanze del centro iperbolico (rappresentazioni concrete, ad alta frequenza), mentre le JEPA di alto livello mappano le rappresentazioni verso la periferia (concetti astratti, piani a lungo orizzonte). Questa struttura geometrica incoraggia la separazione delle scale di astrazione senza la necessità di architetture esplicitamente gerarchiche.

Il calcolo del gradiente in spazio iperbolico richiede la sostituzione del gradiente euclideo con il gradiente riemanniano, che proietta il gradiente euclideo sul corrispondente tangente alla varietà iperbolica. Nella parametrizzazione dell'iperboloide (modello di Lorentz, equivalente ma computazionalmente più stabile):

$\mathcal{H}^n = \{x \in \mathbb{R}^{n+1} : \langle x, x \rangle_{\mathcal{L}} = -1/c, x_0 > 0\}$

con prodotto di Lorentz $\langle x, y \rangle_{\mathcal{L}} = -x_0 y_0 + \sum_{i=1}^{n} x_i y_i$, il gradiente riemanniano $\nabla^{\mathbb{R}} F(x)$ si ottiene proiettando il gradiente euclideo $\nabla F(x)$ sullo spazio tangente $T_x\mathcal{H}^n$:

$T_x\mathcal{H}^n = \{v \in \mathbb{R}^{n+1} : \langle x, v \rangle_{\mathcal{L}} = 0\}$

$\nabla^{\mathbb{R}} F(x) = \nabla F(x) + c \langle x, \nabla F(x) \rangle_{\mathcal{L}} x$

L'operazione di retrazione (proiezione sul manifold dopo un passo del gradiente) si realizza tramite l'esponenziale riemanniana:

$\exp_x(v) = \cosh(\sqrt{c}\, \|v\|_{\mathcal{L}})\, x + \sinh(\sqrt{c}\, \|v\|_{\mathcal{L}})\, v / (\sqrt{c}\, \|v\|_{\mathcal{L}})$

dove $\|v\|_{\mathcal{L}} = \sqrt{\langle v, v \rangle_{\mathcal{L}}}$ è la norma di Lorentz. Gli algoritmi di ottimizzazione riemanniana (Riemannian Adam, Riemannian SGD) sostituiscono l'aggiornamento euclideo $\theta \leftarrow \theta - \eta \nabla F(\theta)$ con $\exp_\theta(-\eta \nabla^{\mathbb{R}} F(\theta))$, garantendo che i parametri rimangano sulla varietà.

6. La curvatura di Ricci e la geometria delle rappresentazioni profonde

Il tensore di curvatura di Riemann misura in modo intrinseco quanto una varietà riemanniana si discosta dall'euclideo: in $\mathbb{R}^n$ il tensore di Riemann si annulla identicamente, mentre la sua traccia, il tensore di Ricci Ric_i□, e la sua traccia ulteriore, la curvatura scalare R, misurano vari aspetti della geometria locale. La connessione tra la curvatura dello spazio delle rappresentazioni e le proprietà statistiche del modello è stata esplorata nel quadro della geometria statistica di Amari.

Per una varietà statistica (Θ, g_F), la curvatura di Ricci si calcola a partire dai simboli di Christoffel della connessione di Levi-Civita. Per le famiglie esponenziali --- come le gaussiane, le categoriali (Gibbs) e le distribuzioni di Dirichlet --- il tensore di curvatura presenta proprietà analitiche note. La famiglia gaussiana a varianza fissa è un'ipersfera di curvatura costante positiva nello spazio delle distribuzioni (dualità e-piatta), mentre la famiglia gaussiana a media e varianza variabili è localmente piatta in una delle due strutture duali di Amari (connessione m-duale). Questa piattezza duale significa che certi problemi di ottimizzazione su queste famiglie hanno una struttura convessa implicita che rende il gradiente naturale esatto e non solo un'approssimazione.

Per le reti neurali profonde come quelle impiegate nei world model, il calcolo esatto del tensore di curvatura è computazionalmente proibitivo. Tuttavia, alcuni indicatori di curvatura scalare hanno trovato applicazioni pratiche: la traccia dell'Hessiano della loss, $\mathrm{tr}(H)$, approssima la curvatura scalare del paesaggio di perdita ed è collegata alla generalizzazione (minima piatta vs. acuta) come si discuterà nella sezione sull'ottimizzazione. La curvatura locale dello spazio delle rappresentazioni è stata studiata empiricamente da Fefferman e collaboratori attraverso il Persistent Homology e la Topological Data Analysis, mostrando che le rappresentazioni di strati profondi di

reti addestrate su ImageNet hanno strutture omologiche complesse che correlano con la struttura semantica del dataset.

7. Trasporto parallelo e apprendimento di metriche invarianti

Il trasporto parallelo è l'operazione che sposta vettori tangenti lungo un cammino su una varietà mantenendoli "costanti" rispetto alla connessione riemanniana. Su una varietà curva, il trasporto parallelo dipende in generale dal cammino scelto, e questa dipendenza dal cammino è il marchio della curvatura: l'olonomia di un anello chiuso misura la rotazione subita da un vettore trasportato parallelamente lungo quel cammino.

Nel contesto dell'apprendimento delle rappresentazioni, il trasporto parallelo è rilevante per la definizione di operazioni di equivarianza nelle reti neurali su varietà. Se si vogliono costruire reti che operino su varietà riemanniane --- come orientazioni di oggetti (SO(3)), configurazioni di articolazioni (varietà di Stiefel o Grassmann), o stati quantistici (sfera di Bloch) --- è necessario definire la convoluzione e l'attenzione in modo che commutino con le isometrie della varietà. Bronstein e collaboratori (2017) nel loro lavoro fondativo sulla geometric deep learning formalizzano questo requisito come equivarianza rispetto al gruppo di isometrie $\mathrm{Isom}(\mathcal{M})$.

Per i world model che operano su sistemi fisici con simmetrie continue --- come bracci robotici soggetti a simmetrie rotazionali o molecole con simmetrie permutazionali --- le architetture equivarianti costruite sulla geometria della varietà degli stati offrono vantaggi di campionamento nell'ordine di diversi ordini di grandezza rispetto alle reti euclidiee non strutturate. La complessità di campionamento di una rete equivariante rispetto a un gruppo G di simmetrie scala come $|G|^{-1}$ rispetto alla rete corrispondente non equivariante, in quanto la rete apprende automaticamente a condividere i parametri tra tutti i punti dell'orbita sotto G.

La connessione con la JEPA è diretta: la parte predittiva dell'architettura JEPA — il modulo che predice la rappresentazione del target dato il contesto — può essere progettata come equivariante rispetto alle trasformazioni rilevanti dello spazio osservativo. Un predittore che opera su sequenze video equivariante rispetto alle traslazioni temporali e alle rotazioni spaziali sarà necessariamente più efficiente nell'estrazione delle strutture causali invarianti dal segnale visivo, e la geometria riemanniana dello spazio delle osservazioni funge da guida per la costruzione di tale equivarianza.

8. Varietà di Wasserstein e geometria ottimale del trasporto

La geometria del trasporto ottimale offre un'ulteriore prospettiva sulla struttura metrica degli spazi di rappresentazioni probabilistiche. Lo spazio di Wasserstein

$W_2(\mathcal{P}(\mathbb{R}^d))$ — lo spazio delle misure di probabilità su $\mathbb{R}^d$ con momento secondo finito, dotato della distanza di Wasserstein di ordine 2 — è esso stesso una varietà riemanniana di dimensione infinita, con proprietà geometriche studiate da Otto (2001) e da Villani (2003, 2009).

La distanza di Wasserstein-2 tra due distribuzioni μ e ν è:

$$W_2^2(\mu, \nu) = \inf_{\gamma \in \Gamma(\mu,\nu)} \iint \|x - y\|^2 \, d\gamma(x, y)$$

dove $\Gamma(\mu, \nu)$ è l'insieme dei piani di trasporto, ossia delle misure su $\mathbb{R}^d \times \mathbb{R}^d$ con marginali μ e ν. Il piano ottimale γ^* è determinato univocamente dal Teorema di Brenier (1991): esiste una funzione convessa φ tale che il trasporto ottimale da μ a ν è la mappa di Brenier $T = \nabla\varphi$ (il gradiente del potenziale di trasporto), e vale $W_2^2(\mu, \nu) = \int \|x - \nabla\varphi(x)\|^2 \, d\mu(x)$.

La struttura riemanniana di Otto su $W_2(\mathcal{P}(\mathbb{R}^d))$ identifica lo spazio tangente in μ con lo spazio delle funzioni $\varphi \in L^2(\mu)$ con gradiente nullo sulla componente ortogonale al supporto di μ, dotato della metrica $\|\varphi\|^2_{T_\mu} = \int \|\nabla\varphi\|^2 \, d\mu$. Le geodetiche nello spazio di Wasserstein sono le interpolazioni di McCann: la distribuzione al tempo t lungo la geodetica tra μ e ν è $(\mathrm{Id} + t(T - \mathrm{Id}))\#\mu$, la spinta in avanti di μ attraverso la mappa lineare tra identità e trasporto ottimale.

Per i world model, la geometria di Wasserstein è rilevante in due contesti distinti. Il primo è la definizione di metriche tra distribuzioni sugli stati latenti: la divergenza KL, usata nell'ELBO, non è una metrica (non soddisfa la simmetria né la disuguaglianza triangolare) e non cattura le proprietà geometriche delle distribuzioni in modo robusto. La distanza W_2 è una vera metrica e ha proprietà di continuità più forti rispetto alla topologia debole delle distribuzioni. Il secondo contesto è l'addestramento avversariale: la distanza di Wasserstein-1 (earth mover's distance) è la funzione di perdita dei modelli GAN di Arjovsky e collaboratori (WGAN, 2017), che ha mostrato proprietà di addestramento più stabili rispetto alla divergenza KL grazie alle sue migliori proprietà geometriche. Per i world model che incorporano componenti generative avversariali, la sostituzione della loss KL con la loss di Wasserstein è motivata geometricamente oltre che empiricamente.

X. Teoria dell'ottimizzazione: paesaggi di loss, mode connectivity e sharpness-aware minimization

1. Il paesaggio di perdita come oggetto geometrico

La funzione di perdita di una rete neurale profonda $L : \mathbb{R}^n \to \mathbb{R}$ definisce una superficie iperdimensionale nello spazio dei parametri che ha proprietà geometriche di straordinaria complessità. L'intuizione visiva di questa superficie come un paesaggio di

montagne, valli e pianure — formulata esplicitamente da Goodfellow, Vinyals e Saxe (2015) e da Li e collaboratori (2018) attraverso tecniche di visualizzazione per proiezione su sottospazi bidimensionali — rivela strutture che hanno conseguenze dirette sulle proprietà di generalizzazione dei modelli addestrati.

Sia $\theta \in \mathbb{R}^n$ il vettore dei parametri del modello. In un punto θ^* della superficie di perdita, la struttura locale è completamente descritta (al secondo ordine) dall'Hessiano:

$H(\theta^*) = \partial^2 L / \partial\theta_i \partial\theta \; |_{\{\theta=\theta^*\}}$

L'Hessiano è una matrice $n \times n$ simmetrica. I suoi autovalori $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda$ determinano le curvature locali nelle direzioni dei corrispondenti autovettori: autovalori grandi positivi corrispondono a direzioni di rapida crescita della loss ("pareti ripide"), autovalori vicini a zero corrispondono a direzioni di variazione lenta ("valli piatte"), e autovalori negativi — se presenti — segnalano punti di sella.

Un minimo locale θ^* è caratterizzato da $\nabla L(\theta^*) = 0$ e $H(\theta^*) > 0$ (Hessiano definito positivo). Un punto di sella soddisfa $\nabla L(\theta^*) = 0$ ma $H(\theta^*)$ ha almeno un autovalore negativo. La ricerca teorica sull'ottimizzazione profonda ha chiarito che il paesaggio di perdita delle reti neurali sovraparametrizzate è dominato da punti di sella, non da minimi locali profondi: la discesa del gradiente stocastica evita generalmente i minimi locali non ottimali perché questi sono estremamente rari in alta dimensione, mentre i punti di sella vengono attraversati grazie alla varianza del gradiente stocastico.

Dauphin e collaboratori (2014) formalizzano questa intuizione osservando che in un paesaggio casuale di dimensione n, la probabilità che un punto critico sia un minimo locale decresce esponenzialmente con n: un punto critico con k autovalori negativi (un punto di sella di indice k) ha probabilità proporzionale a $e^{\{-\alpha k^2/n\}}$ di apparire. Per reti con milioni di parametri, quasi tutti i punti critici sono punti di sella con indice proporzionale a n, e i minimi locali veri hanno tutti valori di loss molto simili al minimo globale.

2. Minimi piatti vs. minimi acuti e il loro legame con la generalizzazione

La distinzione tra minimi piatti (*flat minima*) e minimi acuti (*sharp minima*) nella superficie di perdita è stata indicata per la prima volta da Hochreiter e Schmidhuber (1997) come potenzialmente correlata con la generalizzazione. L'intuizione è che un minimo piatto occupa un grande volume dello spazio dei parametri: piccole perturbazioni di θ^* non aumentano significativamente la loss. Al contrario, un minimo acuto è circondato da pareti ripide, e piccole perturbazioni — come quelle dovute alla varianza del gradiente stocastico o alla differenza tra distribuzione di training e di test — possono portare a un marcato deterioramento delle prestazioni.

Formalmente, si definisce la sharpness di un minimo θ^* rispetto a una perturbazione ε come:

$\varrho(\theta^*, \varepsilon) = \max_{\|\delta\| \leq \varepsilon} [L(\theta^* + \delta) - L(\theta^*)]$

Per un Hessiano con autovalore massimo λ_{max}, il valore massimo di $L(\theta^* + \delta)$ nell'intorno di raggio ε è approssimato da $L(\theta^*) + \varepsilon^2 \lambda_{max} / 2$, da cui $\varrho \approx \varepsilon^2 \lambda_{max} / 2$. Dunque il valore λ_{max} è il misuratore naturale dell'acutezza del minimo.

L'argomento teorico classico di Keskar e collaboratori (2017) mostra empiricamente che la discesa del gradiente con batch di grandi dimensioni (large-batch SGD) converge sistematicamente verso minimi più acuti rispetto alla discesa con piccoli batch, e che i modelli trovati con large-batch generalizzano peggio. La spiegazione è che il grande batch riduce la varianza del gradiente, lasciando la dinamica di ottimizzazione in una traiettoria deterministica che converge al minimo locale più vicino al punto di inizializzazione --- che è spesso un minimo acuto --- invece di permettere l'esplorazione di minimi piatti più lontani ma più benigni.

Tuttavia, l'argomento di Hochreiter-Schmidhuber e Keskar è stato criticato da Dinh e collaboratori (2017) con un'obiezione di carattere fondamentale: la sharpness di un minimo non è invariante rispetto a reparametrizzazioni della rete. Se si sostituisce un peso w con αw e divide il bias corrispondente per α, la funzione realizzata dalla rete non cambia, ma la struttura locale dell'Hessiano --- e quindi la nozione di piattezza --- cambia drasticamente. Questa non-invarianza implica che la sharpness euclidea non è una proprietà intrinseca del modello, ma dipende dalla scelta della parametrizzazione. La risoluzione di questa critica richiede di misurare la sharpness rispetto alla metrica di Fisher (sharpness riemanniana) o di adottare misure invarianti come il volume del bacino di attrazione nel senso della geometria di trasporto ottimale.

3. Sharpness-Aware Minimization: derivazione e analisi

La Sharpness-Aware Minimization (SAM), proposta da Foret, Kleiner, Mobahi e Neyshabur (2021), è un algoritmo di ottimizzazione progettato esplicitamente per trovare minimi piatti minimizzando, invece della loss standard, il limite superiore alla sharpness del minimo trovato. L'obiettivo di SAM è:

$\min_\theta \max_{\|\varepsilon\| \leq \varrho} L(\theta + \varepsilon)$

ossia minimizzare il peggior caso della loss in un intorno di raggio ϱ attorno ai parametri correnti. Questo problema min-max può essere interpretato come la ricerca di parametri θ tali che l'intera palla di raggio ϱ intorno a θ abbia loss bassa, il che è esattamente la caratterizzazione di un minimo piatto.

La difficoltà computazionale sta nel calcolo del massimizzatore interno $\varepsilon^*(\theta) = \operatorname{argmax}_{\|\varepsilon\| \leq \varrho} L(\theta + \varepsilon)$. Poiché questo problema è non convesso in generale, SAM lo approssima con un singolo passo del gradiente ascendente:

$\hat{\varepsilon}(\theta) \approx \varrho \cdot \nabla_\theta L(\theta) / \|\nabla_\theta L(\theta)\|$

che corrisponde alla soluzione esatta dell'approssimazione quadratica locale al primo ordine:

$$\max_{\|\varepsilon\| \leq \rho} L(\theta) + \varepsilon^T \nabla_\theta L(\theta) = L(\theta) + \rho \cdot \|\nabla_\theta L(\theta)\|$$

con $\varepsilon^* = \rho \cdot \nabla_\theta L(\theta) / \|\nabla_\theta L(\theta)\|$. L'aggiornamento SAM si svolge quindi in due passi:

Passo 1 (perturbazione ascendente): $\hat{\varepsilon} = \rho \cdot \nabla_\theta L(\theta_t) / \|\nabla_\theta L(\theta_t)\|$

Passo 2 (discesa del gradiente perturbato): $\theta_{t+1} = \theta_t - \eta \cdot \nabla_\theta L(\theta_t + \hat{\varepsilon})$

Il gradiente nel passo 2 è calcolato nel punto perturbato $\theta_t + \hat{\varepsilon}$, non nel punto originale θ_t. Questo richiede due forward/backward pass per ogni aggiornamento — costo computazionale doppio rispetto a SGD standard — ma la perdita in efficienza è generalmente compensata da miglioramenti nella generalizzazione nell'ordine di 1–3 punti percentuali su benchmark di classificazione di immagini.

La giustificazione teorica di SAM si basa sulla teoria PAC-Bayes. Il limite superiore PAC-Bayes alla generalizzazione di McAllester (1999), applicato alle reti neurali da Dziugaite e Roy (2017), afferma che con probabilità almeno $1 - \delta$ sulla scelta del training set:

$$L_{test}(\theta) \leq L_{train}(P_\theta) + \sqrt{(KL[P_\theta \| P_0] + \log(n/\delta)) / (2n)}$$

dove P_θ è una distribuzione di probabilità sui parametri (versione stocastica del modello), P_0 è un prior sui parametri, e $L_{train}(P_\theta) = \mathbb{E}_{\theta' \sim P_\theta}[L_{train}(\theta')]$ è la loss media sul training set per parametri campionati da P_θ. Se $P_\theta = \mathcal{N}(\theta, \sigma^2 I)$, allora $KL[P_\theta \| P_0]$ è proporzionale alla sharpness del minimo θ e alla norma $\|\theta\|^2$. SAM minimizza implicitamente il termine KL scegliendo minimi θ^* per cui la loss rimane bassa nell'intorno $\mathcal{N}(\theta^*, \rho^2 I)$, il che corrisponde a un basso KL in questo limite PAC-Bayes.

4. Varianti di SAM e adaptive sharpness

La versione originale di SAM utilizza la stessa norma ρ per tutte le componenti del gradiente, ma diverse componenti del vettore dei parametri hanno scale molto diverse in una rete profonda (i parametri dei bias hanno ordini di grandezza diversi dai pesi). L'ASAM (Adaptive SAM, Kwon e collaboratori 2021) introduce una normalizzazione adattiva:

$$\hat{\varepsilon}_{ASAM} = \rho \cdot T(\theta) \nabla_\theta L(\theta) / \|T(\theta) \nabla_\theta L(\theta)\|$$

dove $T(\theta) = \text{diag}(|\theta_1|, |\theta_2|, ..., |\theta|)$ è una matrice diagonale di scala. Questa perturbazione adattiva è invariante rispetto al rescaling dei parametri: se $\theta_i \to \alpha\theta_i$, allora $\hat{\varepsilon}_i \to \alpha\hat{\varepsilon}_i$, così la perturbazione relativa $\hat{\varepsilon}_i/\theta_i$ rimane costante. L'ASAM risolve parzialmente la critica di Dinh, rendendo la sharpness misurata dallo strumento di ottimizzazione più vicina a una grandezza invariante per reparametrizzazione.

La connessione tra SAM e la geometria riemanniana è stabilita da Müller e Zeineldeen (2023): la perturbazione ottimale nel caso della norma di Fisher è $\hat{\varepsilon}_{Fisher} = \rho \cdot F(\theta)^{-1} \nabla_\theta L(\theta) / \|F(\theta)^{-1/2} \nabla_\theta L(\theta)\|$, ossia la perturbazione lungo il gradiente naturale. Questa versione di SAM, denominata Fisher-SAM, minimizza la sharpness riemanniana --- quella invariante per reparametrizzazione --- ed è pertanto immune alla critica di Dinh per costruzione. Il costo computazionale del calcolo di $F(\theta)^{-1}$ rende Fisher-SAM computazionalmente proibitivo per reti grandi, ma approssimazioni KFAC rendono praticabile una versione approssimata.

5. Mode connectivity: struttura topologica del paesaggio di perdita

Uno dei risultati sperimentali più sorprendenti nella teoria dell'ottimizzazione delle reti neurali è il fenomeno della *mode connectivity*: due minimi distinti del paesaggio di perdita possono essere connessi da cammini a bassa perdita nel corrispondente spazio dei parametri. Garipov e collaboratori (2018) hanno scoperto che per reti neurali standard addestrate con SGD, quasi sempre è possibile trovare una curva non rettilinea $\gamma : [0,1] \to \mathbb{R}^n$ con $\gamma(0) = \theta_1$, $\gamma(1) = \theta_2$ tale che $L(\gamma(t))$ sia uniformemente bassa lungo tutta la curva, anche quando $L((1-t)\theta_1 + t\,\theta_2)$ — la loss lungo l'interpolazione lineare — è significativamente più alta.

La curva γ viene parametrizzata come una poligonale di Bézier di grado 2 (curva a gomito) o grado 3, e i parametri della curva vengono ottimizzati minimizzando la loss media lungo la curva:

$$L_{curve}(\gamma) = \int_0^1 L(\gamma(t))\, dt \approx (1/K) \sum L(\gamma(k/K))$$

L'ottimizzazione è rispetto ai parametri del punto interno della poligonale (il "gomito"), tenendo fissi gli estremi θ_1 *e* θ_2.

Risultati simili sono stati ottenuti da Draxler e collaboratori (2018) con un approccio diverso (nudged elastic band), confermando la prevalenza della mode connectivity in architetture diverse. Il fenomeno è particolarmente pronunciato quando i due minimi θ_1 *e* θ_2 sono trovati con condizioni iniziali diverse ma stesso dataset, e diventa meno marcato quando sono trovati con dataset diversi (splitting).

6. La congettura della connettività lineare e il neural tangent kernel

Frankle e collaboratori (2020) hanno avanzato la congettura più forte della *linear mode connectivity*: se due minimi θ_1 *e* θ_2 sono trovati a partire dalla stessa inizializzazione, con lo stesso dataset ma con shuffling diverso dei mini-batch, allora l'interpolazione lineare $L((1-\alpha)\theta_1 + a\,\theta_2)$ ha una barriera di loss (il massimo lungo l'interpolazione meno il minimo) inferiore a una soglia molto piccola (tipicamente < 1% dell'errore di test). Questa congettura è stata verificata empiricamente per reti ResNet e VGG su

CIFAR-10 e ImageNet, e ha implicazioni profonde: significa che i minimi trovati a partire dalla stessa inizializzazione appartengono tutti alla stessa componente connessa del paesaggio di perdita, e che la geometria di questa componente è approssimativamente convessa.

La connessione con il Neural Tangent Kernel (NTK) è essenziale per la comprensione teorica. Il NTK di una rete $f(x; \theta)$ è il kernel definito come:

$$K_{NTK}(x, x') = \langle \nabla_\theta f(x; \theta), \nabla_\theta f(x'; \theta) \rangle = \Sigma_i \, \partial f(x; \theta)/\partial \theta_i \cdot \partial f(x'; \theta)/\partial \theta_i$$

Jacot, Gabriel e Hongler (2018) hanno mostrato che nel limite di larghezza infinita (reti neurali infinitamente larghe), il NTK rimane costante durante il training ($\theta \approx \theta_0 \; \forall \, t$), e la dinamica di training diventa un processo di discesa del gradiente nel feature space del kernel K_{NTK}. In questo regime, il problema di ottimizzazione è convesso nel feature space e la loss converge esponenzialmente a zero. La dinamica lineare implica che tutti i punti del cammino di ottimizzazione hanno loss bassa nel feature space del NTK, il che spiegherebbe la mode connectivity: due reti inizializzate vicine rimangono vicine nello spazio dei gradienti (spazio tangente), e l'interpolazione lineare nel corrispondente regime NTK ha loss bassa.

Per reti di larghezza finita, il NTK non è costante durante il training (regime di feature learning, opposto al regime del kernel), e la dinamica è non lineare. Il regime di feature learning è proprio quello in cui le reti imparano rappresentazioni utili piuttosto che limitarsi a un kernel fisso. La transizione tra regime kernel e regime di feature learning avviene approssimativamente quando la learning rate supera una soglia critica $\eta_c \propto n_width^{-1/2}$ (dove n_width è la larghezza della rete), e in questo regime le proprietà di connettività lineare possono deteriorarsi.

7. L'edge of stability e la dinamica dell'Hessiano durante il training

Un fenomeno empirico osservato da Cohen e collaboratori (2021) con il nome di *edge of stability* (orlo della stabilità) rivela che la discesa del gradiente con learning rate costante η esibisce la seguente dinamica caratteristica: durante le prime epoche del training, il valore dell'autovalore massimo dell'Hessiano λ_{max} cresce monotonamente fino a raggiungere il valore $2/\eta$ — la soglia di stabilità dell'aggiornamento del gradiente per la quale l'aggiornamento $\theta \leftarrow \theta - \eta \, \nabla L$ comincia a oscillare invece di convergere. Raggiunta questa soglia, λ_{max} si stabilizza attorno a $2/\eta$ per il resto del training.

Questo comportamento contraddice la narrativa standard dell'ottimizzazione, secondo la quale la discesa del gradiente funziona bene solo quando $\eta < 2/\lambda_{max}$ (condizione di Lipschitz della loss). L'edge of stability mostra che la dinamica stocastica e l'effetto del progresso dell'ottimizzazione permettono alla rete di operare sistematicamente al limite di questa condizione, dove λ_{max} è tenuto a bada da un meccanismo di

feedback implicito: quando λ_max supera 2/η, l'aggiornamento produce oscillazioni che riducono λ_max, finché non viene ristabilito l'equilibrio.

La teoria dell'edge of stability ha implicazioni dirette per il design degli algoritmi di training dei world model. DreamerV3, ad esempio, adopera una normalizzazione dei gradienti (gradient normalization per layer) che può essere interpretata come un meccanismo di controllo implicito di λ_max: normalizzando i gradienti per strato, si impedisce che gli strati con curvatura locale più alta dominino la dinamica di training e spingano il sistema oltre l'edge of stability.

8. Ottimizzazione Adam e la sua connessione con la geometria della loss

L'algoritmo Adam (Adaptive Moment Estimation, Kingma e Ba 2015) è l'ottimizzatore de facto per l'addestramento di reti neurali profonde, inclusi i world model. La sua formulazione:

$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \nabla_\theta L(\theta_t)$ *(momento del primo ordine)* $v_t = \beta_2 v_{t-1} + (1 - \beta_2) (\nabla_\theta L(\theta_t))^2$ *(momento del secondo ordine)* $\hat{m}_t = m_t / (1 - \beta_1^t)$ *(correzione del bias)* $\hat{v}_t = v_t / (1 - \beta_2^t)$ *(correzione del bias)* $\theta_{t+1} = \theta_t - \eta \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$ (aggiornamento)

dove le operazioni quadratiche e la radice sono element-wise. La divisione per $\sqrt{\hat{v}_t}$ adatta la learning rate a ogni parametro in proporzione inversa alla varianza del gradiente stocastico per quel parametro: parametri con gradienti altamente variabili ricevono learning rate più piccole (avanzano con cautela), mentre parametri con gradienti stabili ricevono learning rate più grandi.

La connessione con la geometria della loss è profonda: Adam approssima l'inversa della matrice di Fisher diagonale, che per distribuzioni gaussiane è $\text{diag}(v_t)^{-1/2}$. L'approssimazione è diagonale e ignora le correlazioni tra parametri diversi, a differenza di KFAC che cattura le correlazioni a livello di layer. Adam è pertanto un gradiente naturale approssimato con approssimazione diagonale della Fisher — una prospettiva che spiega sia i suoi successi che i suoi insuccessi: funziona bene quando la Fisher ha una struttura approssimativamente diagonale, e fallisce quando le correlazioni off-diagonali sono importanti.

Il fenomeno della generalizzazione degradata con Adam rispetto a SGD, osservato da Wilson e collaboratori (2017) su benchmark di classificazione, può essere spiegato in questi termini: Adam, adattando la learning rate per coordinata, tende a fare progressi uniformi lungo tutte le direzioni del parametro, incluse quelle a bassa curvatura che corrispondono a caratteristiche statistiche irrilevanti del training set. SGD, al contrario, avanza preferenzialmente lungo le direzioni di gradiente più grandi — quelle corrispondenti alle caratteristiche più universalmente predittive — e ignora implicitamente le direzioni irrilevanti. Il risultato è che SGD trova minimi più piatti in media, con migliore generalizzazione, ma Adam converge più velocemente dal punto di vista del numero di iterazioni.

9. Catastrophic forgetting, loss landscape e plasticità

Il fenomeno del *catastrophic forgetting* (oblio catastrofico) in apprendimento continuo — la tendenza delle reti neurali a dimenticare rapidamente i task precedenti quando vengono addestrate su nuovi task — ha una spiegazione naturale in termini di paesaggio di perdita. I parametri ottimali per il task precedente θ^*_1 giacciono in una regione dello spazio dei parametri che, dopo il training sul nuovo task, si trova in una regione di alta perdita del nuovo paesaggio: gli aggiornamenti necessari per minimizzare L_2 portano inevitabilmente ad aumentare L_1.

L'algoritmo EWC (Elastic Weight Consolidation, Kirkpatrick e collaboratori 2017) risolve questo problema approssimando localmente l'importanza di ogni parametro per il task precedente attraverso la diagonale della matrice di Fisher F_1:

$$\mathcal{L}_\text{EWC}(\theta) = L_2(\theta) + (\lambda/2)\, \Sigma_i\, F_1_ii\, (\theta_i - \theta^*_1_i)^2$$

Il termine di regolarizzazione penalizza lo spostamento dai parametri ottimali θ_1 *in proporzione all'importanza di ogni parametro per il task precedente (misurata da* F_1_ii*). L'interpretazione geometrica è che EWC costruisce un potenziale quadratico attorno a* θ_1 che simula l'effetto di un prior gaussiano bayesiano sui parametri: $\theta \sim \mathcal{N}(\theta^*_1, (\lambda F_1)^{-1})$.

Per i world model che devono apprendere continuamente da nuove osservazioni senza dimenticare le strutture dinamiche già apprese — il problema del *continual world model learning* — la connessione tra EWC e geometria della Fisher offre un punto di partenza formale. La struttura a moduli separati (encoder, modello di transizione, decoder) permette di applicare EWC selettivamente al solo modulo di transizione, preservando la conoscenza dinamica mentre il modulo percettivo si adatta alle nuove condizioni visive.

Il problema della *plasticità* nella sua forma più recente, studiato da Nikishin e collaboratori (2022) nel contesto del RL profondo, mostra che reti neurali addestrate per periodi lunghi in ambienti non stazionari perdono progressivamente la capacità di apprendere nuove informazioni — un fenomeno opposto all'oblio catastrofico ma parimenti problematico. La perdita di plasticità è correlata con l'accumulo di parametri morti (neuroni con attivazioni costantemente nulle) e con il deterioramento della condizione numerica delle matrici di Gram dello spazio latente. Il paesaggio di perdita di una rete che ha perso plasticità è caratterizzato da un'elevata densità di punti di sella degeneri (con molteplici autovalori nulli dell'Hessiano), nei quali la discesa del gradiente non riesce a trovare direzioni di discesa sufficientemente informative.

10. Connessione con i world model: loss landscape della predizione latente

Per la JEPA e i world model in generale, la struttura del paesaggio di perdita presenta caratteristiche specifiche che meritano un'analisi separata. La loss di predizione latente:

$\mathcal{L}_{pred} = \mathbb{E}_{x,y}[D(h_\varphi(f_\theta(x), z), g_\psi(y))]$

dove h_φ è il predittore, f_θ e g_ψ sono encoder di contesto e target, e z è un token posizionale, ha un paesaggio di perdita con due categorie distinte di punti critici problematici.

La prima categoria è quella dei *collassi di rappresentazione*, ossia soluzioni degeneri in cui f_θ e g_ψ mappano tutti gli input allo stesso vettore costante (collasso totale) o a vettori in un sottospazio di dimensione bassa (collasso parziale). In questi minimi, la loss di predizione si annulla perché il predittore impara a predire il vettore costante, ma le rappresentazioni non contengono informazione sull'input. L'Hessiano in questi punti è semidefinito positivo con un sottospazio di kernel di dimensione proporzionale alla dimensione dello spazio latente non utilizzato — sono minimi degeneri nel senso che ogni direzione che preserva il collasso non aumenta la loss.

La seconda categoria è quella dei *falsi attrattori di alta curvatura*, ossia minimi dove le rappresentazioni codificano caratteristiche superficiali dell'input (texture, colori medi) invece delle strutture semanticamente rilevanti. Questi minimi hanno curvatura elevata (λ_{max} grande) nelle direzioni corrispondenti a perturbazioni semantiche dell'input, e curvatura bassa nelle direzioni corrispondenti a variazioni superficiali. SAM, operando nell'intorno di raggio ϱ, dovrebbe in principio sfuggire a questi minimi spostandosi verso direzioni di minore acutezza — ma il raggio ϱ deve essere calibrato con cura per essere sufficientemente grande da attraversare le barriere tra il minimo spurio e il bacino del minimo semantico.

Le strategie anti-collasso della JEPA — stop-gradient, momentum encoder (EMA), regolarizzazione della varianza-covarianza — modificano il paesaggio di perdita in modo da eliminare i collassi come soluzioni: lo stop-gradient rompe la simmetria del collasso totale introducendo un'asimmetria tra i due branch dell'encoder, mentre la regolarizzazione della varianza aggiunge barriere di perdita attorno ai collassi parziali. Dal punto di vista della topologia del paesaggio, queste tecniche trasformano i minimi degeneri da punti di minima loss a punti di sella o soluzioni di alta loss, forzando l'ottimizzazione a convergere verso rappresentazioni non degeneri.

XI. Meta-learning e adattamento rapido del modello del mondo

1. Il problema del meta-apprendimento: apprendere a imparare

L'apprendimento automatico convenzionale richiede grandi quantità di dati per ogni nuovo compito: un modello del mondo addestrato sulla dinamica di un robot in un ambiente specifico non possiede, in linea di principio, alcun vantaggio rispetto a un modello inizializzato casualmente quando viene confrontato con un nuovo robot in un nuovo ambiente. Questa fragilità contrasta con la capacità degli agenti biologici di adattarsi a situazioni radicalmente nuove dopo poche esperienze, sfruttando strutture apprese in contesti precedenti. Il meta-apprendimento — o apprendimento del secondo ordine — si propone di colmare questo divario costruendo algoritmi che imparino non solo a svolgere un compito specifico, ma a imparare qualsiasi compito proveniente da una distribuzione di compiti affini con una quantità di dati drasticamente ridotta.

La formalizzazione del meta-apprendimento richiede di distinguere tra due livelli di apprendimento. Il livello esterno (*outer loop*, o meta-training) ottimizza un *meta-parametro* Θ attraverso esposizione a molti compiti campionati da una distribuzione $p(\mathcal{T})$. Il livello interno (*inner loop*, o task-specific adaptation) usa Θ come punto di partenza per adattarsi rapidamente a ogni singolo compito $\mathcal{T} \sim p(\mathcal{T})$ con un piccolo numero di passi di ottimizzazione o di esempi. La qualità del meta-parametro Θ si misura dalla prestazione attesa dopo adattamento veloce:

$$\mathcal{L}_\text{meta}(\Theta) = \mathbb{E}_{\mathcal{T} \sim p(\mathcal{T})} \left[\mathcal{L}_{\mathcal{T}}(A(\Theta, \mathcal{D}_{\mathcal{T}}\{\text{support}\}))\right]$$

dove $A(\Theta, \mathcal{D}_{\mathcal{T}}\{\text{support}\})$ è il parametro adattato al compito $\mathcal{T}$ usando il support set $\mathcal{D}_{\mathcal{T}}\{\text{support}\}$ (pochi esempi), e $\mathcal{L}_{\mathcal{T}}$ è valutata sul query set $\mathcal{D}_{\mathcal{T}}\{\text{query}\}$ del medesimo compito. La separazione in support e query set simula la situazione di test: il modello vede pochi esempi del nuovo compito e viene valutato su altri esempi dello stesso compito.

Questa formulazione è deliberatamente agnostica rispetto alla forma di A: la funzione di adattamento può essere un numero fisso di passi di discesa del gradiente (MAML), un'operazione di attenzione che legge dalla memoria (prototypical networks, matching networks), o un'intera rete ricorrente che implementa l'ottimizzazione come computazione interna (meta-SGD, optimization meta-learning). Ciascuna di queste scelte porta a proprietà computazionali e statistiche diverse che si manifestano in modo cruciale quando il meta-apprendimento viene applicato ai modelli del mondo.

2. MAML: derivazione dell'algoritmo e del gradiente di secondo ordine

L'algoritmo MAML (Model-Agnostic Meta-Learning) di Finn, Abbeel e Levine (2017) è la formulazione più influente del meta-apprendimento basato su gradiente. La sua idea centrale è che i meta-parametri Θ possano essere inizializzati in modo tale che un piccolo numero di passi di discesa del gradiente sufficienti per adattarsi a qualsiasi compito della distribuzione $p(\mathcal{T})$.

Sia f_θ un modello con parametri θ. Per ogni compito $\mathcal{T}$ campionato da p($\mathcal{T}$), MAML esegue K passi di discesa del gradiente a partire da Θ:

$$\theta'_{\mathcal{T}} = \Theta - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}}^{train}(\theta) \,|\, \{\theta=\Theta\} \quad \text{(per K = 1)}$$

$$\theta'^{(k)}_{\mathcal{T}} = \theta'^{(k-1)}_{\mathcal{T}} - \alpha \nabla_\theta \mathcal{L}_{\mathcal{T}}^{train}(\theta'^{(k-1)}_{\mathcal{T}}) \quad \text{(per K passi)}$$

La meta-ottimizzazione aggiorna Θ minimizzando la loss sul query set dopo adattamento:

$$\Theta \leftarrow \Theta - \beta \nabla_\Theta \Sigma_{\mathcal{T}} \mathcal{L}_{\mathcal{T}}^{test}(\theta'_{\mathcal{T}})$$

Il calcolo del gradiente $\nabla_\Theta \mathcal{L}_{\mathcal{T}}^{test}(\theta'_{\mathcal{T}})$ richiede di differenziare attraverso l'operazione di adattamento, ossia attraverso i K passi della discesa del gradiente interna. Per K = 1, questo produce un gradiente del secondo ordine:

$$\partial \mathcal{L}_{\mathcal{T}}^{test}(\theta'_{\mathcal{T}}) / \partial\Theta = \partial \mathcal{L}_{\mathcal{T}}^{test} / \partial\theta'_{\mathcal{T}} \cdot \partial\theta'_{\mathcal{T}} / \partial\Theta$$

$$= \partial \mathcal{L}_{\mathcal{T}}^{test} / \partial\theta'_{\mathcal{T}} \cdot (I - \alpha \partial^2 \mathcal{L}_{\mathcal{T}}^{train} / \partial\theta\partial\theta^T)$$

Il termine $\partial^2 \mathcal{L}_{\mathcal{T}}^{train} / \partial\theta\partial\theta^T$ è l'Hessiano della loss di training rispetto ai parametri — il costo computazionale dominante di MAML. Il calcolo esatto di questo Hessiano ha complessità $O(n^2)$ in memoria e $O(n^3)$ in tempo per n parametri, il che lo rende impraticabile per reti profonde moderne. La variante FOMAML (First-Order MAML) ignora questo termine ($\partial^2 \mathcal{L} / \partial\theta^2 \approx 0$), approssimando il gradiente meta con il solo primo termine:

$$\nabla_\Theta^{FOMAML} \mathcal{L}_{\mathcal{T}}^{test} \approx \partial \mathcal{L}_{\mathcal{T}}^{test} / \partial\theta'_{\mathcal{T}}$$

Sorprendentemente, FOMAML mantiene la maggior parte dei benefici empirici di MAML a una frazione del costo computazionale. La spiegazione teorica è che il termine di secondo ordine contribuisce principalmente alla direzione del gradiente quando l'adattamento produce cambiamenti grandi nei parametri, mentre per adattamenti piccoli (α piccolo, K = 1) il termine di secondo ordine è trascurabile in norma.

L'algoritmo Reptile di Nichol e Schulman (2018) fornisce un'alternativa ancora più semplice: invece di calcolare gradienti attraverso l'adattamento, Reptile aggiorna Θ verso i parametri adattati $\theta'_{\mathcal{T}}$:

$$\Theta \leftarrow \Theta + \varepsilon (\theta'_{\mathcal{T}} - \Theta)$$

Nichol, Achiam e Schulman (2018) mostrano che Reptile è equivalente a FOMAML più un termine di regolarizzazione che massimizza la similarità tra i gradienti di training di compiti diversi campionati dall'inizio della traiettoria di ottimizzazione interna. L'interpretazione geometrica è che Reptile muove Θ verso un punto che sia simultaneamente vicino agli ottimi di molti compiti — il centro di gravità dei bacini di attrazione dei diversi compiti nella metrica euclidea dei parametri.

3. Meta-apprendimento bayesiano e incertezza sul compito

L'approccio bayesiano al meta-apprendimento tratta Θ come un prior sullo spazio dei parametri e l'adattamento come inferenza bayesiana della posteriore del task dato il support set. Il framework PLATIPUS (Finn e collaboratori 2018) parametrizza $\Theta = (\mu_\Theta, \Sigma_\Theta)$ come media e covarianza di una distribuzione gaussiana sui parametri del task:

$$p(\theta \mid \Theta) = \mathcal{N}(\theta; \mu_\Theta, \Sigma_\Theta)$$

Dopo osservazione del support set $\mathcal{D}_{\mathcal{T}}\{\text{support}\}$, l'adattamento bayesiano produce la posteriore:

$$p(\theta \mid \mathcal{D}_{\mathcal{T}}\{\text{support}\}, \Theta) \propto p(\mathcal{D}_{\mathcal{T}}\{\text{support}\} \mid \theta)\, p(\theta \mid \Theta)$$

$$\approx \mathcal{N}(\theta; \mu_{\{\mathcal{T}\}}, \Sigma_{\{\mathcal{T}\}})$$

La garanzia dell'incertezza sulla stima del compito è cruciale per i world model applicati a domini robotici: quando il support set è estremamente ridotto (1–5 traiettorie), l'incertezza sull'identità del compito corrente può essere grande, e un sistema che ignori questa incertezza potrebbe pianificare in modo troppo ottimistico. Il framework bayesiano produce distribuzioni sulle previsioni del world model:

$$p(s_{t+1} \mid s_t, a_t, \mathcal{D}^{\{\text{support}\}}) = \int p_\theta(s_{t+1} \mid s_t, a_t)\, p(\theta \mid \mathcal{D}^{\{\text{support}\}}, \Theta)\, d\theta$$

che catturano sia l'incertezza aleatoria intrinseca (varianza di p_θ dato θ) sia l'incertezza epistemica sul compito (varianza rispetto alla posteriore su θ). Questa distinzione, discussa nel capitolo sui POMDP, diventa computazionalmente trattabile nel framework meta-bayesiano perché la posteriore è gaussiana e l'integrale ha forma analitica o una buona approssimazione variazionale.

4. MAML applicato ai modelli del mondo: meta-world models

L'applicazione di MAML ai modelli del mondo richiede di specificare cosa costituisce un "compito" nella distribuzione $p(\mathcal{T})$. Nella robotica, i compiti naturali includono: diverse morfologie del robot (differente numero di giunti, diversa distribuzione di massa), diversi parametri fisici dell'ambiente (attrito, elasticità, rigidità dei materiali), diversi obiettivi del task (afferrare diversi oggetti, navigare in ambienti con diversa topologia). La distribuzione $p(\mathcal{T})$ codifica la variabilità attesa nell'applicazione.

Clavera e collaboratori (2018) introducono MAML applicato specificamente all'adattamento dei modelli dinamici in robotica, con un inner loop che esegue pochi passi di discesa del gradiente sul modello di transizione usando traiettorie recenti, e un

outer loop che ottimizza i meta-parametri attraverso compiti di dinamica diversa. Formalmente, il modello di transizione è $f_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$, e il meta-apprendimento ottimizza:

$$\Theta = \text{argmin}_\Theta \, \mathbb{E}_{\mathcal{T}}[\Sigma_t \| f_{\theta'_{\mathcal{T}}}(s_t, a_t) - s_{t+1}\|^2]$$

dove $\theta'_{\mathcal{T}} = \Theta - \alpha \nabla_\theta [\Sigma_t \| f_\theta(s_t, a_t) - s_{t+1}\|^2] \, | \, \{\theta=\Theta\}$

Il risultato principale è che un robot con meta-parametri Θ addestrati su una distribuzione di masse e attriti diversi riesce ad adattarsi a una nuova configurazione fisica dopo appena una traiettoria di 200 passi, riducendo l'errore di predizione del modello di transizione di un fattore 5–10 rispetto a un modello statico non adattativo.

Nagabandi e collaboratori (2019) estendono questo approccio in MAML-P, dove il meta-apprendimento è combinato con un meccanismo di context encoding: oltre ai parametri del modello $\theta'_{\mathcal{T}}$, il sistema mantiene un vettore di contesto $c_{\mathcal{T}} \in \mathbb{R}^d$ che codifica le caratteristiche del compito corrente e condiziona il modello di transizione:

$$f_{\theta, c}(s_t, a_t) = \text{MLP}_\theta([s_t; a_t; c_{\mathcal{T}}])$$

Il vettore di contesto $c_{\mathcal{T}}$ viene aggiornato con un encoder ricorrente applicato alle traiettorie recenti — un meccanismo che permette un adattamento più rapido del solo gradiente interno perché il contesto può essere aggiornato online senza retropropagazione.

5. Modelli del mondo con inferenza del task: LILAC e VariBAD

Una linea di ricerca distinta ma complementare al meta-apprendimento basato su gradiente affronta il problema dell'adattamento come un problema di inferenza bayesiana sull'identità del task. Se i parametri fisici del compito corrente $\mathcal{T}$ (ad esempio, massa del robot, attrito del pavimento) possono essere parametrizzati da un vettore $\omega \in \Omega$, allora l'agente dovrebbe mantenere una belief state $b_t(\omega) = p(\omega \mid \mathcal{D}_{1:t})$ che si aggiorna a ogni nuova osservazione usando la regola di Bayes:

$$b_t(\omega) \propto p(o_t, r_t \mid s_t, a_t, \omega) \cdot b_{t-1}(\omega)$$

Il modello del mondo condizionato su ω produce previsioni ottimali per la fisica corrente:

$$p(s_{t+1} \mid s_t, a_t, b_t) = \int p_\omega(s_{t+1} \mid s_t, a_t) \, b_t(\omega) \, d\omega$$

L'algoritmo VariBAD (Zintgraf e collaboratori 2020) approssima questa belief bayesiana con un encoder variazionale: una rete ricorrente che processa la storia delle transizioni $(s_{1:t}, a_{1:t}, r_{1:t})$ e produce i parametri di una distribuzione gaussiana su un vettore latente $m_t \in \mathbb{R}^d$ che approssima l'identità del task:

$q_\varphi(m \mid \mathcal{D}_{1:t}) = \mathcal{N}(m; \mu_\varphi(\mathcal{D}_{1:t}), \sigma^2_\varphi(\mathcal{D}_{1:t}))$

La politica condiziona sulle statistiche di questa distribuzione: $\pi_\omega(a_t \mid s_t, \mu_t, \sigma_t)$. L'addestramento end-to-end ottimizza l'ELBO sulla likelihood delle traiettorie condizionate al vettore latente del task, in modo analogo all'RSSM ma con la variabile latente che codifica le caratteristiche del task piuttosto che lo stato del mondo.

LILAC (Efficient Meta-Reinforcement Learning via Online Context Variables, 2020) combina l'inferenza del task con il model-based RL: il vettore di contesto m_t viene aggiornato online usando l'encoder variazionale, e il modello del mondo condizionato su m_t viene usato per pianificazione con Dreamer. Il risultato è che il sistema si adatta alla fisica del compito corrente entro le prime 5–10 transizioni reali, mentre la politica pianifica immaginariamente nel modello condizionato al contesto corrente.

6. Algoritmi evolutivi per il meta-apprendimento dei world model

Un approccio alternativo al meta-apprendimento basato sul gradiente è il meta-apprendimento evolutivo, che ottimizza i meta-parametri Θ attraverso algoritmi evolutivi invece della retropropagazione. Questo approccio è particolarmente rilevante per i modelli del mondo perché evita il problema del calcolo degli Hessiani di secondo ordine e gestisce naturalmente obiettivi non differenziabili come il ritorno atteso sul compito.

Clune e collaboratori (2019) propongono Meta-Learning via Evolutionary Algorithms (META), dove la distribuzione $p(\mathcal{T})$ è progettata automaticamente attraverso un processo di curriculum evolutivo: i compiti più difficili (che producono i maggiori gradienti evolutivi) vengono selezionati dinamicamente durante il meta-training, e la difficoltà del curriculum aumenta progressivamente.

La strategia evolutiva CMA-ES (Covariance Matrix Adaptation Evolution Strategy), già impiegata nel lavoro originale di Ha e Schmidhuber per ottimizzare il controllore del world model, è particolarmente adatta come outer loop del meta-apprendimento per world model di piccola dimensione: CMA-ES mantiene una distribuzione gaussiana $\mathcal{N}(\mu, C)$ sui meta-parametri e aggiorna la media μ e la matrice di covarianza C in base alla fitness dei campioni:

$\mu \leftarrow \Sigma_i \, w_i \, \theta_i$ (aggiornamento della media pesato per fitness)

$C \leftarrow (1 - c_{cov}) \, C + c_{cov} \, \Sigma_i \, w_i \, (\theta_i - \mu)(\theta_i - \mu)^\top$ (aggiornamento della covarianza)

dove $\{\theta_i\}$ sono i campioni dall'iterazione corrente e $\{w_i\}$ sono i pesi di fitness (i migliori campioni hanno peso maggiore). La matrice di covarianza C cattura la struttura di correlazione dello spazio dei meta-parametri nel bacino di attrazione del meta-ottimo corrente, permettendo passi di ottimizzazione adattati alla geometria locale.

7. In-context learning come meta-apprendimento implicito nei Transformer

Una delle scoperte più sorprendenti dell'era dei grandi modelli linguistici è che i Transformer di dimensione sufficiente esibiscono *in-context learning* (ICL): dati k esempi input-output nel prompt, il modello esegue un task nuovo senza aggiornare i propri pesi, semplicemente condizionando le previsioni sugli esempi forniti. Dal punto di vista del meta-apprendimento, l'ICL è un'implementazione implicita dell'inner loop: la "funzione di adattamento" è realizzata dall'attenzione del Transformer, che pesa i token del prompt in funzione della loro pertinenza per la query corrente.

Akyürek e collaboratori (2022) e Von Oswald e collaboratori (2023) dimostrano teoricamente che un Transformer è in grado di implementare un passo di discesa del gradiente come computazione di attenzione: per dati lineari ($y = Wx + \varepsilon$), l'attenzione multi-testa di un singolo Transformer layer implementa esattamente un passo di aggiornamento ai minimi quadrati. Formalmente, dato il prompt $[(x_1, y_1), ..., (x , y), x_e \square\square]$, l'attenzione produce:

$\hat{y} = x_e \square\square^T (\Sigma_i x_i x_i^T)^{-1} (\Sigma_i x_i y_i) = x_e \square\square^T W_LS$

dove W_LS è la soluzione ai minimi quadrati del problema di regressione definito dagli esempi nel contesto. Questo è esattamente FOMAML con un solo passo gradient per la regressione lineare.

Per i modelli del mondo, la prospettiva dell'ICL come meta-apprendimento ha implicazioni concrete: un world model implementato come Transformer autoregressivo su sequenze di (s_t, a_t, r_t, s_{t+1}) può potenzialmente eseguire adattamento rapido alla fisica del compito corrente semplicemente condizionando le previsioni sulla storia recente nel context window, senza necessità di aggiornamento dei parametri. Hafner (2023) sperimenta in questa direzione con DreamerV3, osservando che aumentando la lunghezza del context window il modello migliora la sua capacità di adattarsi a variazioni di dinamica all'interno di una singola traiettoria.

Il limite fondamentale dell'ICL come meta-apprendimento è la lunghezza finita del contesto: un Transformer con context window W può adattarsi a compiti che richiedono al massimo W esempi per l'adattamento, il che esclude compiti che richiedono molte traiettorie per la stima dei parametri fisici. Le architetture con memoria esterna (Transformer-XL, Mamba, modelli con external memory store) estendono questo limite comprimendo la storia in un vettore di stato che può essere trasportato attraverso episodi di lunghezza arbitraria.

8. Curriculum learning e distribuzione di compiti adattativi

La qualità del meta-apprendimento dipende criticamente dalla distribuzione di compiti $p(\mathcal{T})$ su cui viene condotto il meta-training. Una distribuzione troppo stretta (pochi compiti simili) porta a meta-parametri overfit che non generalizzano a compiti nuovi; una distribuzione troppo ampia (compiti troppo diversi) rende il problema di meta-apprendimento impossibile perché nessuna inizializzazione comune è adatta a tutti i compiti. Il curriculum learning adattativo modifica $p(\mathcal{T})$ dinamicamente durante il meta-training in risposta alle prestazioni correnti del meta-learner.

Portelas e collaboratori (2020) propongono ALP-GMM (Absolute Learning Progress Gaussian Mixture Model), un algoritmo che traccia il progresso di apprendimento su ogni compito e campiona preferenzialmente compiti su cui il progresso è massimo:

$$\text{score}(\mathcal{T}) = |L_{\mathcal{T}}(t) - L_{\mathcal{T}}(t - \Delta t)| \; / \; \Delta t$$

La distribuzione di campionamento aggiornata è proporzionale a $\text{score}(\mathcal{T})$: i compiti troppo facili (progresso = 0) e troppo difficili (progresso = 0) vengono selezionati raramente, mentre i compiti alla frontiera delle capacità correnti vengono selezionati frequentemente. Questo meccanismo implementa il principio di Vygotsky della zona di sviluppo prossimale in forma algoritmica.

Per i meta-world model, il curriculum adattativo può essere costruito sull'errore di predizione del modello di transizione: i compiti per cui il modello ha l'errore di predizione più alto — ossia quelli dove la dinamica appresa è meno accurata — vengono campionati con più frequenza nel meta-training, guidando l'apprendimento verso le regioni dello spazio dei compiti meno coperte. La metrica di promozione del campionamento diventa:

$$\text{priority}(\mathcal{T}) = \mathbb{E}_{(s,a,s') \sim \mathcal{T}} \left[\| f_\Theta(s, a) - s' \|^2 \right]$$

con un termine di smoothing per evitare di focalizzarsi su compiti eccezionalmente difficili che non appartengono all'effettiva distribuzione di test.

9. Meta-apprendimento delle funzioni di ricompensa e inverse RL

Un aspetto spesso trascurato nel meta-apprendimento dei modelli del mondo è l'adattamento non solo della dinamica (modello di transizione) ma anche degli obiettivi (funzione di ricompensa). L'apprendimento per rinforzo inverso (IRL, Inverse Reinforcement Learning) ricostruisce la funzione di ricompensa di un agente a partire dalle sue traiettorie di comportamento osservate, con l'assunzione che l'agente sia ottimale rispetto a quella funzione. Il meta-IRL combina IRL con meta-apprendimento per adattare rapidamente la funzione di ricompensa a un nuovo task osservando poche dimostrazioni.

L'algoritmo MAML-IRL (Finn e collaboratori 2017, sezione robotica) esegue il loop interno sul problema IRL — adattando la funzione di ricompensa attraverso

MaxEntropy IRL — e il loop esterno meta-ottimizza i parametri iniziali della funzione di ricompensa:

$r_\varphi(s, a) \rightarrow r_{\{\varphi'_\mathcal{T}\}}(s, a)$ (inner loop: IRL su dimostrazioni del task $\mathcal{T}$)

$\Phi \leftarrow \Phi - \beta \nabla_\Phi \mathbb{E}_{\{\mathcal{T}\}}[J_\mathcal{T}(\pi^*, r_{\{\varphi'_\mathcal{T}\}})]$ (outer loop: meta-ottimizzazione)

dove $J_\mathcal{T}(\pi^*, r)$ è il ritorno atteso della politica ottimale rispetto alla funzione di ricompensa r nel task $\mathcal{T}$. Il sistema risultante è in grado di inferire la ricompensa di un nuovo task da 1–3 dimostrazioni e di pianificare immediatamente con il world model adattato.

La combinazione di meta-world model (modello di transizione adattativo) e meta-IRL (funzione di ricompensa adattativa) produce un sistema capace di adattamento multi-componente: dato un nuovo task, il sistema inferisce simultaneamente la fisica dell'ambiente e l'obiettivo dell'agente dalle traiettorie osservate, producendo un modello del mondo completamente adattato per pianificazione e controllo. Questa capacità corrisponde a una forma di teoria della mente computazionale applicata ai sistemi fisici: riconoscere non solo come il mondo si muove, ma perché si muove (intenzionalità).

10. Limiti teorici del meta-apprendimento e il problema dell'overfitting sul meta-training

La teoria della generalizzazione per il meta-apprendimento è meno sviluppata di quella per l'apprendimento ordinario, ma alcuni risultati fondamentali delimitano i possibili benefici. Baxter (2000) e Maurer (2009) mostrano che la complessità di generalizzazione del meta-apprendimento scala con $\sqrt{(C/m + C_{task}/n)}$ dove m è il numero di compiti nel meta-training, n è il numero di esempi per compito, C e C_{task} sono complessità dipendenti dalla classe dell'ipotesi meta e task. Questo risultato implica che il meta-apprendimento è benefico solo quando la distribuzione di compiti è sufficiently structured (bassa complessità effettiva dell'insieme dei task) e quando il numero di compiti m è abbastanza grande.

Per i world model robotici, il problema del meta-overfitting è particolarmente insidioso: se la distribuzione di training $p(\mathcal{T})$ include solo robot con braccia e ambienti planari, il meta-world model apprende una funzione di inizializzazione ottimizzata per questo spazio, che potrebbe non trasferirsi a robot con strutture radicalmente diverse o a ambienti tridimensionali. La soluzione pratica è massimizzare la diversità del meta-training set attraverso domain randomization (randomizzazione di massa, attrito, lunghezze dei link, posizioni degli oggetti), ma i limiti statistici di Maurer-Baxter suggeriscono che la diversità ha un costo in termini di efficienza di campionamento sul singolo task. Il compromesso ottimale tra diversità e densità dei compiti rimane un problema aperto, sia teoricamente che empiricamente.

XII. Apprendimento causale e rappresentazioni causali nei modelli del mondo

1. Il limite della correlazione statistica e la necessità di strutture causali

La critica più profonda ai modelli del mondo neurali contemporanei non riguarda le loro architetture specifiche né le procedure di ottimizzazione, bensì il fondamento epistemologico dell'apprendimento da dati osservazionali. Una rete neurale addestrata a massimizzare la verosimiglianza predittiva su una distribuzione di traiettorie apprende correlazioni statistiche tra stati, azioni e osservazioni successive — e le sfrutta per produrre previsioni accurate all'interno della distribuzione di addestramento. Ma correlazione e causalità sono oggetti matematicamente distinti, e la differenza tra loro diventa determinante non appena il sistema viene interrogato su situazioni che implicano interventi, controfattuali, o generalizzazione fuori distribuzione.

La distinzione fu formalizzata con rigore matematico da Judea Pearl a partire dagli anni Novanta nella teoria dei grafi causali e del calcolo-do (*do-calculus*). Pearl distingue tre livelli di un'ipotetica gerarchia causale, denominata *Ladder of Causation*: il primo livello è quello dell'associazione (correlazione statistica, predizione passiva), formalizzato dalla distribuzione congiunta $p(x, y)$; il secondo è quello dell'intervento, formalizzato dalla distribuzione interventistica $p(y \mid do(x = v))$ che descrive cosa accade a y quando x è forzato al valore v indipendentemente dalle sue cause naturali; il terzo è quello del controfattuale, che risponde a domande del tipo "cosa sarebbe successo a y se x fosse stato v, dato che in realtà ho osservato $x = u$ e $y = w$?". Ogni livello superiore è strettamente più ricco del precedente: non è possibile in generale calcolare distribuzioni interventistiche a partire da sole statistiche associazionali, né distribuzioni controfattuali a partire da sole distribuzioni interventistiche.

Per un world model, operare al livello dell'associazione significa costruire un simulatore statisticamente accurato dell'esperienza passata ma privo di comprensione delle cause. Un tale simulatore fallirà sistematicamente quando l'agente pianifica azioni che modificano la struttura causale dell'ambiente — che è esattamente ciò che ogni agente deve fare per agire efficacemente nel mondo.

2. Grafi causali diretti aciclici e il framework strutturale di Pearl

Un Grafo Causale Diretto Aciclico (DAG, Directed Acyclic Graph) è una coppia $(\mathcal{V}, \mathcal{E})$ dove $\mathcal{V} = \{X_1, ..., X\}$ è l'insieme delle variabili e $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ è l'insieme degli archi direzionati senza cicli. Un arco $X_i \rightarrow X$ indica che X_i è causa diretta di X . I genitori

di una variabile X sono i nodi con un arco diretto verso di essa: $pa(X) = \{X_i : X_i \rightarrow X \in \mathcal{E}\}$.

Il Modello Strutturale Causale (Structural Causal Model, SCM) associato a un DAG specifica, per ogni variabile X, un'equazione strutturale:

$X = f(pa(X), N)$

dove f è una funzione deterministica e N è un termine di rumore esogeno, indipendente da tutti gli altri rumori $\{N_i\}_{i \neq}$. Le equazioni strutturali codificano i meccanismi causali: ciascuna f è un meccanismo autonomo che non cambia quando si modificano le altre equazioni — questa proprietà di invarianza modulare è il marchio distintivo delle relazioni causali rispetto alle correlazioni statistiche.

La distribuzione osservazionale $p(X_1, ..., X)$ fattorizza secondo la struttura del DAG:

$p(X_1, ..., X) = \Pi_{j=1}^{n} p(X \mid pa(X))$

Questa fattorizzazione, nota come *Markov condition*, è valida per qualsiasi distribuzione generata da un SCM. L'intervento $do(X = v)$ produce una nuova distribuzione interventistica ottenuta rimuovendo l'equazione strutturale per X e sostituendola con $X = v$ (recidendo tutti gli archi entranti in X nel DAG):

$p(X_{-j} \mid do(X = v)) = \Pi_{i \neq j} p(X_i \mid pa(X_i)) |_{X = v}$

Il *do-calculus* di Pearl (1995, 2000) fornisce un sistema di tre regole di inferenza che permette di calcolare distribuzioni interventistiche a partire dalla distribuzione osservazionale e dal grafo causale, quando il grafo è noto. Le tre regole governano rispettivamente: l'inserzione/rimozione di osservazioni, l'interscambio di interventi e osservazioni, e l'inserzione/rimozione di interventi. Pearl dimostrò la completezza del do-calculus: se una distribuzione interventistica è identificabile dalla distribuzione osservazionale, il do-calculus la calcola.

3. Identificabilità causale e il problema dell'apprendimento della struttura

Il problema centrale dell'apprendimento causale dai dati è che un dataset osservazionale finito è in generale compatibile con molte strutture causali diverse — l'insieme dei DAG compatibili con le statistiche di ordine superiore osservate è detto *classe di equivalenza di Markov* (o equivalence class di Markov), caratterizzata graficamente dal Pattern Essential (CPDAG, Completed Partially Directed Acyclic Graph). Solo gli interventi sperimentali, che rompono la struttura osservazionale variando attivamente le variabili, permettono di distinguere tra DAG nella stessa classe di Markov.

Gli algoritmi di apprendimento della struttura causale si dividono in due famiglie principali. Gli algoritmi *constraint-based* (PC, FCI, RFCI) sfruttano i test di indipendenza

condizionale: se $X_i \perp\!\!\!\perp X \mid Z$ in p, allora nel DAG corretto non esiste un cammino attivo (d-separated path) tra X_i e X dato Z. L'algoritmo PC costruisce il CPDAG iterativamente: inizia con un grafo completo non diretto, rimuove gli archi corrispondenti a coppie con indipendenza condizionale, e orienta gli archi rimanenti usando le regole di Meek (1995). La complessità computazionale è O(n^{k+2}) test di indipendenza, dove k è il grado massimo di separazione, con n il numero di variabili.

Gli algoritmi *score-based* (GES, NOTEARS, DAG-GNN) ottimizzano uno score bayesiano o di informazione mutua sulla struttura del grafo. NOTEARS (No Tears, Zheng e collaboratori 2018) trasforma il problema di apprendimento del DAG da un problema di ricerca combinatoria su strutture discrete a un problema di ottimizzazione continua, sfruttando la caratterizzazione:

un grafo $W \in \mathbb{R}^{n\times n}$ è un DAG $\Leftrightarrow h(W) = \mathrm{tr}(e^{W \circ W}) - n = 0$

dove $\circ$ è il prodotto di Hadamard e l'esponenziale è della matrice. Questo vincolo, equivalente all'aciclicità, può essere incorporato in un problema di ottimizzazione penalizzata:

$\min_W (1/2n)\|X - XW^T\|_F^2 + \lambda\|W\|_1 + \varrho\, h(W)^2$

La formulazione continua di NOTEARS permette di usare algoritmi di ottimizzazione differenziabile (L-BFGS, gradient descent), rendendo il problema trattabile per grafi con decine di variabili. La sua estensione a modelli non lineari (NOTEARS-MLP) sostituisce la regressione lineare con MLP parametrizzate per ogni arco del grafo.

4. Rappresentazioni causali disaccoppiate: il programma di Schölkopf

Bernhard Schölkopf e collaboratori (2021, "Toward Causal Representation Learning") hanno delineato il programma della *causal representation learning* come la fusione tra l'apprendimento profondo delle rappresentazioni e la teoria causale di Pearl. Il punto di partenza è che i dati sensoriali grezzi — pixel di immagini, sequenze sonore, segnali tattili — non sono le variabili causalmente rilevanti: sono proiezioni ad alta dimensione di variabili latenti causali di basso livello (posizione degli oggetti, proprietà dei materiali, identità delle entità) che interagiscono secondo relazioni causali semplici nello spazio latente.

L'obiettivo della causal representation learning è apprendere simultaneamente le variabili causali latenti $\{Z_1, ..., Z\}$ e la struttura causale del DAG che le governa, a partire da dati osservazionali grezzi. Questo problema è fondamentalmente più difficile dell'apprendimento causale su variabili osservate: si tratta di una forma di identificazione non lineare di componenti indipendenti (non-linear ICA) con vincoli causali.

Il principio di *Independent Causal Mechanisms* (ICM) di Peters, Schölkopf e Janzing (2017) fornisce un criterio per distinguere la direzione causale: tra due variabili correlate X e

Y, la direzione causale corretta X → Y è quella per cui p(Y | X) e p(X) sono "più indipendenti" come oggetti informativi — ossia, conoscere la distribuzione marginale di X non fornisce informazioni sulla distribuzione condizionale di Y dato X. Formalmente:

$$I(p(X) ; p(Y|X)) \leq I(p(Y) ; p(X|Y))$$

dove I è la informazione mutua tra descrizioni della distribuzione. Questa disuguaglianza può essere stimata attraverso la lunghezza descrittiva minima dei due fattori della fattorizzazione.

5. Modelli del mondo strutturalmente causali: CausalDreamer e varianti

L'integrazione del framework causale negli RSSM e nei world model di tipo Dreamer ha prodotto alcune architetture recenti di notevole interesse. CausalDreamer (Guo e collaboratori 2023) modifica l'RSSM inserendo una struttura a variabili latenti causalmente disaccoppiate: invece di un singolo vettore $z_t \in \mathbb{R}^d$ per lo stato stocastico, il modello usa un insieme di variabili $\{Z_{,}\}_{k=1}^K$, ciascuna di dimensione ridotta, con una struttura di dipendenza appresa in modo sparso.

La fattorizzazione causale dello stato stocastico è:

$$p(Z_{1,}, ..., Z_{,} \mid h_t) = \Pi_{k=1}^{K} p(Z_{,} \mid pa(Z_{,}), h_t)$$

dove $pa(Z_{,}) \subseteq \{Z_{1,-1}, ..., Z_{,-1}\} \cup h_t$ sono i genitori causali di $Z_{,}$ nel DAG latente. La struttura sparsa del DAG latente è appresa attraverso un meccanismo di attenzione che produce una matrice di pesi $A \in [0,1]$ dove $A \approx 0$ indica assenza di dipendenza causale da Z verso $Z_{,}$ e $A \approx 1$ indica dipendenza forte. La sparsità è incoraggiata da una penalità L_1:

$$\mathcal{L}_{sparse} = \lambda \Sigma_{k,l} A$$

Questa struttura causale disaccoppiata produce rappresentazioni latenti con le seguenti proprietà verificabili: interventi su una singola variabile latente Z propagano i loro effetti solo attraverso i discendenti di Z nel DAG, lasciando invariate le variabili non discendenti (indipendenza causale modulare). Questa proprietà può essere testata empiricamente generando rollout immaginari con e senza interventi su Z e verificando che le variabili non discendenti non cambino distribuzione.

Il vantaggio pratico di questa struttura è la generalizzazione a interventi fuori distribuzione: il modello del mondo causale predice correttamente le conseguenze di azioni che modificano selettivamente alcune variabili latenti, mentre un RSSM standard monolotica fallisce perché non ha mai visto nella distribuzione di training l'effetto di quel tipo di intervento parziale.

Un'estensione di particolare rilievo per la tesi di questa monografia è C-JEPA (Causal Joint Embedding Predictive Architecture, Volokhova e collaboratori, 2024), che porta il paradigma della predizione latente mascherata dal livello di patch al livello di oggetto. Mentre I-JEPA maschera e predice regioni rettangolari dell'immagine indipendentemente dalla struttura semantica della scena, C-JEPA individua le entità oggettuali mediante un meccanismo di slot attention e opera il masking a livello di interi oggetti — occultando il token che rappresenta un oggetto specifico e chiedendo al predittore di inferirne la rappresentazione dalla scena rimanente. Su benchmark di ragionamento controfattuale in ambienti fisici strutturati come CLEVRER, C-JEPA ottiene un guadagno assoluto di circa il 20% rispetto alla stessa architettura con masking a livello di patch, confermando empiricamente che le rappresentazioni object-centric supportano il ragionamento causale in modo qualitativamente superiore alle rappresentazioni patch-based.

Sul piano teorico, C-JEPA formalizza il masking a livello di oggetto come un'operazione di intervento latente: rimuovere la rappresentazione di un oggetto equivale a eseguire un do(Z_obj = ∅) sul grafo causale dello stato, recidendo gli archi entranti nel nodo oggetto e forzando il predittore a inferire lo stato dell'oggetto dai soli effetti che esso produce sugli altri oggetti della scena. L'analisi teorica di C-JEPA introduce il concetto di vicinato di influenza (influence neighborhood) di un oggetto k come l'insieme degli oggetti la cui distribuzione latente cambia significativamente quando Z_k viene mascherato; questo concetto è più debole dell'identificazione causale piena richiesta dal do-calculus di Pearl, ma è operativamente calcolabile in spazi ad alta dimensionalità e fornisce una misura pratica della struttura causale che il modello ha appreso senza supervisione esplicita sulle relazioni tra oggetti. La nozione di vicinato di influenza si configura così come alternativa operativa alla causal discovery esplicita, particolarmente adatta ai regimi in cui lo spazio degli stati ha dimensionalità troppo elevata per l'applicazione diretta degli algoritmi di apprendimento della struttura come NOTEARS o PCMCI.

Una risposta complementare alle difficoltà dei pianificatori a campionamento è offerta da GRASP (*Gradient RelAxed Stochastic Planner*, cfr. *Parallel Stochastic Gradient-Based Planning for World Models* di Psenka e collaboratori, 2026), che affronta il problema dal lato opposto: invece di ridurre la dimensionalità delle rappresentazioni, migliora l'efficienza dell'ottimizzazione per modelli del mondo visivi ad alta dimensionalità. Il contributo centrale di GRASP è la separazione tra stati e azioni come variabili di ottimizzazione distinte, ispirata alle tecniche di *multiple shooting* del controllo ottimo classico: invece di propagare serialmente gli stati attraverso il modello del mondo (come in MPPI e CEM), GRASP tratta gli stati intermedi come "stati virtuali" ottimizzati in parallelo, eliminando la necessità di rollout seriali e riducendo drasticamente il condizionamento del problema.

Per gestire la geometria avversariale degli spazi latenti visivi — dove i gradienti rispetto agli input di stato tendono a essere *brittle* e sfruttabili — GRASP introduce un meccanismo di *stop-gradient* sugli input di stato del modello del mondo, conservando solo i gradienti rispetto alle azioni. L'esplorazione è garantita da aggiornamenti di tipo Langevin sugli stati virtuali, che permettono di sfuggire ai minimi locali attraverso

perturbazioni stocastiche controllate. Sui benchmark *long-horizon* in spazi visivi, GRASP supera CEM e *gradient descent* standard sia in tasso di successo che in velocità di convergenza, con guadagni fino al 10% sul tasso di successo a meno della metà del costo computazionale.

La giustificazione teorica è rigorosa: gli autori dimostrano che il metodo di *shooting* seriale ha una costante di Lipschitz del gradiente che cresce esponenzialmente con l'orizzonte di pianificazione, mentre l'approccio *lifted* mantiene una costante uniforme indipendente dall'orizzonte — esattamente il tipo di vantaggio scalabile che i world model per la robotica incarnata richiedono.

6. Il do-calculus per la pianificazione interventistica

La pianificazione ottimale in un modello del mondo strutturalmente causale richiede il calcolo di distribuzioni interventistiche, non osservazionali. L'agente non osserva passivamente il mondo: agisce su di esso, il che equivale a eseguire do(A = a) nel linguaggio di Pearl. La differenza tra la distribuzione osservazionale p(S +1 | S , A) e la distribuzione interventistica p(S +1 | do(A = a)) è rilevante quando ci sono variabili di confondimento (*confounders*) — variabili non osservate che influenzano sia l'azione sia lo stato.

In un sistema di controllo robotico, i confounders possono essere stati interni dell'agente (fatica, calibrazione dei sensori) che influenzano sia le azioni eseguite sia le osservazioni ricevute. La formula del backdoor di Pearl permette di calcolare la distribuzione interventistica da dati osservazionali quando è disponibile un insieme di aggiustamento Z (un insieme che blocca tutti i percorsi di backdoor dal confounders all'effetto):

p(S +1 | do(A = a)) = Σ_z p(S +1 | A = a , Z = z) p(Z = z)

In un world model neurale, la formula del backdoor può essere implementata marginando sull'encoder del confounders: se il vettore di stato latente z_t approssima il confounder, la distribuzione interventistica si ottiene mediando le previsioni del modello di transizione su z_t campionato dalla distribuzione marginale del prior, invece che dalla distribuzione condizionale al contesto corrente.

La connessione con l'inferenza variazionale dell'RSSM è diretta: la formula del backdoor corrisponde all'uso del prior p(z_t | h_t) invece della posteriore q(z_t | h_t, o_t) nel calcolo delle previsioni. Mentre la posteriore incorpora informazione sull'osservazione corrente (inclusi i possibili effetti dei confounders), il prior margina sui confounders. Questa interpretazione giustifica la distinzione tra prior e posterior nell'RSSM non solo dal punto di vista variazionale, ma anche causale: il prior implementa il do-calculus, la posterior implementa il condizionamento osservazionale.

7. Controfattuali nei world model: immaginazione e ragionamento ipotetico

I controfattuali sono richieste di ragionamento del tipo: "cosa sarebbe successo se avessi eseguito azione a' invece di a, dato che ho eseguito a e ho osservato o?" Questa forma di ragionamento, che Pearl colloca al terzo livello della gerarchia causale, è fondamentale per l'apprendimento dall'esperienza — la capacità di imparare da errori e successi passati immaginando corsi di azione alternativi.

La formula controfattuale nel framework SCM di Pearl si calcola in tre passi (abduction-action-prediction):

Passo 1 (Abduction): inferire la distribuzione dei rumori esogeni N dato l'osservazione: p(N | X = x_obs)

Passo 2 (Action): modificare il SCM per implementare l'intervento: X_j ← v invece di f (pa(X), N)

Passo 3 (Prediction): calcolare la distribuzione dell'outcome nel SCM modificato con i rumori inferiti: Y_{counterfactual} = f_Y(pa(Y), N_Y)

Nel contesto dei world model neurali, la realizzazione del Passo 1 corrisponde all'inferenza della posteriore sull'osservazione corrente — esattamente quello che la componente encoder dell'RSSM esegue. Il Passo 2 corrisponde alla sostituzione dell'azione nella sequenza di rollout. Il Passo 3 corrisponde al rollout immaginario del world model con i rumori inferiti e l'azione controfattuale.

Buesing e collaboratori (2019) mostrano come questa struttura possa essere implementata esplicitamente in un modello del mondo neurale con variabili di rumore esplicite: mantenendo i valori dei rumori esogeni N_t fissi tra la traiettoria osservata e quella controfattuale, il modello produce scenari alternativi coerenti — che differiscono solo per le azioni eseguite, non per il contesto stocastico sottostante. Questa coerenza è essenziale per produrre controfattuali sensati invece di traiettorie spurie che cambiano non solo in risposta all'azione diversa ma anche in risposta a realizzazioni diverse del rumore.

8. Causalità e transfer learning: il principio di invarianza

Una delle connessioni più fertili tra teoria causale e apprendimento automatico riguarda il problema del domain shift: un modello addestrato su una distribuzione di dati p(X, Y) può fallire su un'altra distribuzione q(X, Y) perché sfrutta correlazioni statistiche che sono presenti in p ma non in q. Il framework causale offre una spiegazione unificata di questo fenomeno e una soluzione principiata.

Peters, Bühlmann e Meinshausen (2016) introducono il principio di *Invariant Causal Prediction* (ICP): tra tutte le possibili rappresentazioni $S \subseteq X$ delle feature di input, quella che utilizza solo i genitori causali diretti di Y nel DAG è la rappresentazione più stabile rispetto a interventi (distribuzioni diverse). Formalmente, per qualsiasi distribuzione interventistica p^e(X, Y) generata da un intervento sull'ambiente e:

$$p^e(Y \mid X_S) = p^{e'}(Y \mid X_S) \ \forall \ e, e' \in \mathcal{E} \Leftrightarrow S = pa(Y)$$

Il metodo ICP stima i genitori causali di Y testando l'invarianza della distribuzione condizionale p(Y | X_S) attraverso ambienti diversi (raccolte di dati da condizioni diverse): si cerca il più piccolo S per cui la condizione di invarianza non viene rifiutata dai test statistici.

Arjovsky e collaboratori (2019) propongono *Invariant Risk Minimization* (IRM), un'estensione al setting delle reti neurali: invece di testare l'invarianza dopo aver trovato la rappresentazione, IRM include l'invarianza come vincolo nell'ottimizzazione:

$$\min_{\Phi, w} \Sigma_e \mathcal{R}^e(w \circ \Phi) + \lambda \| \nabla_w \mathcal{R}^e(w \circ \Phi) | \{w=1\} \|^2$$

dove Φ è l'encoder, w è il classificatore lineare, $\mathcal{R}^e$ è il rischio empirico nell'ambiente e, e il secondo termine penalizza la dipendenza del gradiente ottimale dall'ambiente. L'idea è che se la rappresentazione Φ cattura le feature causalmente invarianti, il classificatore ottimale w non dipende dall'ambiente e il gradiente è zero per ogni ambiente.

Per i world model, IRM suggerisce di addestrare il modello di transizione in modo da minimizzare l'errore di predizione simultaneamente su più ambienti con dinamiche diverse, penalizzando le rappresentazioni che richiedono decoder diversi per ambienti diversi. La rappresentazione causalmente invariante è quella per cui un singolo decoder funziona uniformemente bene su tutti gli ambienti — corrispondente alla struttura causale invariante tra ambienti.

9. Causalità temporale e causal discovery nelle sequenze

Nei world model, le variabili di interesse sono tipicamente sequenze temporali: stati s_1, s_2, ..., s , azioni a_1, ..., a_{-1}. La struttura causale nelle sequenze temporali ha caratteristiche proprie, formalizzate nel concetto di *causalità di Granger* e nella sua estensione strutturale.

La causalità di Granger (1969) afferma che una serie temporale X causa in senso Granger Y se la conoscenza della storia di X riduce l'errore di predizione di Y:

$$Var[Y \mid Y_{<t}, X_{<t}] < Var[Y \mid Y_{<t}]$$

Questa nozione è statistica (non strutturale nel senso di Pearl) e non distingue tra causalità diretta e mediata da variabili latenti. Peters e collaboratori (2013) introducono la *Causalità di Granger strutturale* che richiede inoltre che la relazione sia stabile rispetto a interventi — avvicinandola alla causalità nel senso di Pearl.

Per le sequenze di stati latenti di un world model, la scoperta della struttura causale temporale può essere implementata come problema di apprendimento della struttura del DAG sulle variabili latenti in finestre temporali adiacenti. PCMCI (Runge e collaboratori 2019) estende l'algoritmo PC alle serie temporali prendendo in considerazione dipendenze ritardate: per ogni coppia (X_i, $-\tau$, X ,) con ritardo $\tau = 0$, ..., τ_max, il test condiziona su un insieme di "condizioni di Markov" che blocca i percorsi spurii attraverso la storia.

L'applicazione di PCMCI alle rappresentazioni latenti di un RSSM addestrato su dati di sistemi fisici (Ke e collaboratori 2021) rivela che le variabili latenti apprese con architetture standard hanno spesso struttura di dipendenza densa (ogni variabile dipende da tutte le altre), mentre le variabili latenti di modelli strutturati causalmente (CausalDreamer) mostrano strutture sparse che corrispondono meglio alle dipendenze fisiche reali. Questa corrispondenza tra struttura latente appresa e struttura causale fisica è un indicatore empirico della qualità del grounding causale del modello.

10. Disentanglement e causalità: l'impossibilità del non-supervisionato

Un risultato fondamentale di Locatello e collaboratori (2019), denominato *impossibility theorem for unsupervised disentanglement*, stabilisce che senza supposizioni induttive appropriate, nessun metodo non supervisionato può identificare variabili latenti disaccoppiate in modo univoco: esistono sempre infinite famiglie di rappresentazioni equivalenti (nel senso della verosimiglianza) che differiscono per la struttura di accoppiamento tra variabili latenti.

Formalmente, il teorema di Locatello afferma che per qualsiasi metodo di apprendimento delle rappresentazioni che ottimizzi una loss che dipende solo dalla distribuzione di generazione p(x) (e non dall'identità delle variabili latenti), esiste una biiezione continua non-triviale delle variabili latenti che produce la stessa distribuzione osservazionale. Questa biiezione in generale non preserva il disaccoppiamento: variabili latenti apparentemente disaccoppiate rispetto alla rappresentazione originale possono essere accoppiate nella rappresentazione trasformata.

Il teorema di Locatello non esclude il disaccoppiamento, ma richiede che venga introdotta informazione aggiuntiva: etichette supervisionate (anche poche), supervisione debole attraverso gruppi di osservazioni prodotte con lo stesso valore di alcune variabili latenti, o vincoli strutturali motivati da considerazioni causali. Nel framework causale, i meccanismi f delle equazioni strutturali forniscono esattamente questo tipo di vincoli: il principio ICM afferma che meccanismi diversi sono

indipendenti, il che implica che la rappresentazione causalmente corretta è quella per cui p(Z | pa(Z)) è il meccanismo più "semplice" per ogni j.

Per la JEPA, il teorema di Locatello impone un limite fondamentale: la predizione latente non supervisionata non può, in linea di principio, produrre rappresentazioni causalmente disaccoppiate senza vincoli aggiuntivi. Le strategie pratiche per approssimare il disaccoppiamento causale includono: l'uso di diversità di punti di vista (augmentations che implementano interventi parziali), l'addestramento su sequenze di azioni esplicite (che corrispondono a interventi reali sull'ambiente), e l'uso di vincoli architetturali che implementano la modularità dei meccanismi (slot attention, object-centric representation). Nessuna di queste strategie è equivalente a un approccio causale rigoroso, ma insieme producono rappresentazioni con migliori proprietà di trasferimento causale rispetto alle reti monolitiche.

XIII. Teoria dell'attenzione e dei Transformer per i modelli del mondo

1. Origini e motivazione del meccanismo di attenzione

L'emergenza dei meccanismi di attenzione nell'apprendimento automatico non fu il frutto di un'unica intuizione isolata, ma la convergenza di più filoni di ricerca che, a partire dai primi anni Duemiladieci, riconoscevano un limite strutturale delle architetture ricorrenti classiche. Le reti LSTM e GRU, pur avendo mitigato il problema del gradiente che svanisce attraverso meccanismi di gating, soffrivano di una dipendenza sequenziale intrinseca: il calcolo dello stato al passo t non poteva cominciare prima di aver completato quello al passo $t - 1$, il che rendeva la parallelizzazione su hardware moderno sostanzialmente impossibile durante il training su sequenze lunghe. Al contempo, il trasporto dell'informazione rilevante attraverso centinaia di passi temporali imponeva all'informazione un percorso attraverso un collo di bottiglia — il vettore di stato nascosto — con perdita progressiva e sistematica delle dipendenze a lungo raggio.

Il meccanismo di attenzione, nella sua formulazione archetipica di Bahdanau, Cho e Bengio (2015), propose un rimedio radicale: invece di costringere l'encoder a comprimere l'intera sequenza di input in un singolo vettore di contesto, permettere al decoder di accedere direttamente a tutti gli stati nascosti dell'encoder attraverso un peso di rilevanza appreso dinamicamente. Questo peso esprimeva, per ogni posizione del decoder, quali posizioni dell'encoder fossero pertinenti per la generazione del token corrente. La conseguenza fu immediata in termini di qualità nei sistemi di traduzione automatica, ma la portata teorica si estese ben oltre quella prima applicazione: l'attenzione aveva reso esplicita la struttura delle dipendenze tra elementi di una sequenza, rimuovendo il vincolo dell'intermediazione attraverso stati latenti fissi.

Vaswani e collaboratori (2017), nell'articolo "Attention Is All You Need", portarono questa intuizione alla sua conseguenza più radicale: eliminare completamente la ricorrenza e la convoluzione, costruendo un'architettura — il Transformer — basata esclusivamente su meccanismi di attenzione e trasformazioni posizione per posizione. L'architettura risultante fu parallelizzabile in modo massiccio, scalò in modo eccezionale con il compute, e dimostrò empiricamente di catturare dipendenze a lungo raggio con un'efficienza senza precedenti. La sua adozione nei modelli del mondo, dalle architetture TWM a IRIS, da GENIE ai video diffusion model con backbone a Transformer, è la manifestazione di questa superiorità strutturale.

2. Derivazione scalata del self-attention: il meccanismo di base

Si consideri una sequenza di n vettori di input $x_1, \ldots, x \in \mathbb{R}^d$, organizzati come righe della matrice $X \in \mathbb{R}^{nxd}$. Il meccanismo di self-attention produce per ogni posizione i una rappresentazione contestualizzata $\tilde{y}_i$ che incorpora informazioni da tutte le altre posizioni, con pesi determinati dalla pertinenza reciproca dei contenuti.

Il primo passo consiste nel proiettare X in tre spazi distinti attraverso matrici di parametri apprese $W_Q \in \mathbb{R}^{dxd}$□, $W_K \in \mathbb{R}^{dxd}$□, $W_V \in \mathbb{R}^{dxd_v}$:

$Q = X\ W_Q \in \mathbb{R}^{nxd}$□ (query) $K = X\ W_K \in \mathbb{R}^{nxd}$□ (chiave) $V = X\ W_V \in \mathbb{R}^{nxd_v}$ (valore)

Le query codificano ciò che ogni posizione sta cercando; le chiavi codificano ciò che ogni posizione offre; i valori codificano il contenuto effettivo che viene aggregato. I punteggi di compatibilità tra la query alla posizione i e la chiave alla posizione j sono calcolati come prodotto scalare:

e_i□ $= q_i^T\ k\ / \sqrt{d}$

Il fattore di normalizzazione $1/\sqrt{d}$ è cruciale. Senza di esso, per valori grandi di d il prodotto scalare assume valori con magnitudine proporzionale a d (poiché per vettori gaussiani isotropi $\mathbb{E}[q^Tk] = 0$ e $\mathrm{Var}[q^Tk] = d$). Punteggi grandi spingono la softmax verso regioni di gradiente quasi nullo — la regione di saturazione — deteriorando significativamente il flusso del gradiente durante il training. La divisione per $\sqrt{d}$ mantiene la varianza dei punteggi dell'ordine dell'unità indipendentemente dalla dimensione.

La distribuzione di attenzione per la posizione i è la distribuzione di probabilità normalizzata sui punteggi:

α_i□ $= \mathrm{softmax}(e_{i1}, \ldots, e_i$□$) = \exp(e_i$□$) / \Sigma\ \exp(e_i$□$)$

L'uscita della posizione i è la combinazione convessa dei valori pesata dalle probabilità di attenzione:

$\tilde{y}_i = \Sigma \ \alpha_i \square \ v$

In forma matriciale compatta, l'intera operazione di attenzione scalata si scrive:

$\text{Attn}(Q, K, V) = \text{softmax}(QK^T / \sqrt{d}) \ V$

dove la softmax è applicata riga per riga alla matrice dei punteggi $QK^T/\sqrt{d} \in \mathbb{R}^{n \times n}$, producendo la matrice degli pesi di attenzione $A \in \mathbb{R}^{n \times n}$ con $A_i \square = \alpha_i \square$ e $\Sigma \ A_i \square = 1$ per ogni i. Il costo computazionale dell'attenzione scalata è $O(n^2 d)$ in tempo e $O(n^2)$ in spazio per la memorizzazione della matrice A — la dipendenza quadratica dalla lunghezza della sequenza è il principale collo di bottiglia per sequenze molto lunghe, come le sequenze di patch video ad alta risoluzione nei modelli del mondo visivi.

3. Analisi spettrale dell'attenzione: connessione con i kernel

La forma dell'attenzione scalata può essere analizzata attraverso la lente della teoria dei kernel. La matrice dei punteggi normalizzata $\exp(QK^T/\sqrt{d})$ è proporzionale a un kernel esponenziale applicato ai prodotti scalari tra query e chiavi. Se si definisce il feature map $\varphi(x) = \exp(\|x\|^2/2) \ [\exp(\sqrt{2} \ x_1/\sqrt{d}), \ldots, \exp(\sqrt{2} \ x /\sqrt{d})]$ (come nelle espansioni di Bochner per kernel isotropi positivi definiti), la valutazione dell'attenzione è riconducibile a un'operazione di convoluzione nel feature space.

Questa connessione con la teoria dei kernel spiega alcune proprietà qualitative del meccanismo. Il prodotto scalare $q^T k$ misura la similarità lineare tra la query e la chiave; la trasformazione esponenziale rende la distribuzione di attenzione sensibile a differenze anche piccole tra punteggi (la softmax amplifica gli scarti). Per query e chiavi su cui è stata applicata la normalizzazione di layer (layer normalization), il prodotto scalare è proporzionale al coseno dell'angolo tra i vettori proiettati, rendendo l'attenzione invariante alla norma delle rappresentazioni — una proprietà che contribuisce alla stabilità numerica del training.

Katharopoulos e collaboratori (2020) sfruttano questa struttura kernelizzata per costruire la *linear attention*: approssimando $\exp(q^T k/\sqrt{d}) \approx \varphi(q)^T \varphi(k)$ con funzioni kernel positive, si può riscrivere l'attenzione come:

$\tilde{y}_i = \Sigma \ \varphi(q_i)^T \varphi(k) \ v \ / \ \Sigma \ \varphi(q_i)^T \varphi(k) = \varphi(q_i)^T \ (\Sigma \ \varphi(k) \ v^T) \ / \ \varphi(q_i)^T \ (\Sigma \ \varphi(k))$

Poiché il termine $\Sigma \ \varphi(k) \ v^T \in \mathbb{R}^{d\varphi \times d_v}$ e il termine $\Sigma \ \varphi(k) \in \mathbb{R}^{d\varphi}$ possono essere precalcolati una volta sola per tutte le posizioni, la complessità in tempo dell'attenzione lineare diventa $O(n \ d^{\varphi} \ d_v)$ — lineare in n invece di quadratica. La qualità dell'approssimazione dipende dalla scelta del feature map φ; la funzione $\varphi(x) = \text{elu}(x) + 1$ (con elu la funzione di attivazione ELU) garantisce la positività dei pesi di attenzione. La linear attention è particolarmente rilevante per i world model su sequenze video molto lunghe (migliaia di token per clip di minuti), dove la quadraticità standard è computazionalmente proibitiva.

4. Multi-head attention: decomposizione per sottospazi

L'attenzione a singola testa calcola un insieme di pesi di rilevanza che opera sull'intero spazio d . Questa scelta ha una limitazione fondamentale: i pattern di dipendenza rilevanti per un task complesso possono essere di natura eterogenea — alcune posizioni sono importanti per la struttura sintattica, altre per la coreferenza semantica, altre ancora per le relazioni causali. Un'unica distribuzione di attenzione non può rappresentare simultaneamente pattern di dipendenza multipli e qualitativamente diversi.

L'attenzione multi-testa (multi-head attention, MHA) risolve questo problema attraverso la decomposizione in h teste parallele, ciascuna operante su un sottospazio diverso delle rappresentazioni:

head = Attn(X W^H_Q, X W^H_K, X W^H_V)

MHA(X) = Concat(head$_1$, …, head) W_o

dove W^H_Q ∈ $\mathbb{R}^{dxd}$□/ , W^H_K ∈ $\mathbb{R}^{dxd}$□/ , W^H_V ∈ $\mathbb{R}^{dxd_v}$/ sono le matrici di proiezione specifiche per la testa h, e W_o ∈ $\mathbb{R}^{d_v xd}$ è la matrice di proiezione di output. Scegliendo d = d_v = d/h (dimensioni ridotte proporzionalmente), il costo computazionale dell'MHA è equivalente a quello di un'attenzione a testa singola di dimensione piena — h teste più piccole costano quanto una testa grande.

Il beneficio è che ogni testa può specializzarsi su un tipo diverso di dipendenza. Analisi empiriche delle teste di attenzione in Transformer addestrati su linguaggio mostrano specializzazioni spontanee: alcune teste catturano relazioni sintattiche locali (dipendenza verbo-soggetto), altre catturano dipendenze a lungo raggio (coreferenza tra pronomi e antecedenti lontani), altre ancora implementano pattern quasi-posizionali (dipendenze con offset fisso). Nei modelli del mondo visivi, analoghe specializzazioni emergono: alcune teste catturano correlazioni spaziali locali (bordi, texture), altre codificano la struttura globale dell'immagine (sfondo vs. foreground), altre tracciano le traiettorie di oggetti attraverso il tempo.

La proiezione finale W_o combina le rappresentazioni di tutte le teste in uno spazio unificato, permettendo all'informazione aggregata da diversi pattern di dipendenza di interagire nei layer successivi.

5. Positional encoding: codifica della struttura sequenziale

Il meccanismo di self-attention è intrinsecamente invariante alla permutazione: se si rimescolano le righe di X, l'output viene rimescolato nello stesso modo, senza che il contenuto di ciascuna rappresentazione cambi (poiché il prodotto scalare tra ogni coppia di posizioni rimane identico). Questa proprietà di equivarianza alla

permutazione è vantaggiosa in alcuni contesti (insiemi non ordinati, grafi), ma inadeguata per sequenze in cui l'ordine è semanticamente rilevante — il testo, la sequenza temporale di frame video, la sequenza spaziale di patch in un'immagine.

La codifica posizionale (*positional encoding*) aggiunge a ciascun embedding di input un vettore che dipende dalla posizione:

$\tilde{x}_i = x_i + PE(i)$

dove $PE(i) \in \mathbb{R}^d$ è il vettore di codifica posizionale per la posizione i. Vaswani e collaboratori propongono una codifica sinusoidale con frequenze multiple:

PE(i, 2k) = sin(i / 10000^(2k/d)) PE(i, 2k+1) = cos(i / 10000^(2k/d))

dove $k = 0, 1, \ldots, d/2 - 1$ indica il canale. Questa scelta ha proprietà notevoli. La differenza $PE(i+\delta) - PE(i)$ è una funzione periodica di i e δ, il che garantisce che il modello possa imparare a ragionare sulle distanze relative. In particolare, è possibile esprimere $PE(i+\delta)$ come trasformazione lineare di $PE(i)$:

PE(i+δ, 2k) = PE(i, 2k) cos(δ/ω) + PE(i, 2k+1) sin(δ/ω) PE(i+δ, 2k+1) = −PE(i, 2k) sin(δ/ω) + PE(i, 2k+1) cos(δ/ω)

con ω = 10000^(2k/d). Questa relazione lineare permette al modello di identificare distanze relative attraverso prodotti scalari tra codifiche posizionali. Le frequenze basse (k piccolo) variano lentamente con i e codificano la posizione assoluta a grande scala; le frequenze alte (k grande) variano rapidamente e codificano la struttura fine.

I Transformer moderni per i world model visivi impiegano spesso codifiche posizionali apprese (*learned positional embeddings*): $PE(i) = p_i \in \mathbb{R}^d$ con p_i un vettore di parametri addestrati per backpropagation. Questa scelta sacrifica la generalizzazione a sequenze di lunghezza arbitraria ma offre maggiore flessibilità rappresentativa. Una generalizzazione importante è il *RoPE* (Rotary Position Embedding, Su e collaboratori 2021), che codifica la posizione come rotazione del vettore di query e chiave nel piano complesso:

R q_i = [q_2□_-1 cos(mθ) − q_2□ sin(mθ), q_2□_-1 sin(mθ) + q_2□ cos(mθ)]_ =1^{d/2}

dove $\theta = 1/10000^{(2k/d)}$ e m è la posizione assoluta. Il prodotto scalare tra la query alla posizione m e la chiave alla posizione n è allora:

(R q)^T (R k) = Re[Σ (q_2□_-1 + i q_2□) (k_2□_-1 − i k_2□) exp(i(m−n)θ)]

che dipende solo dalla distanza relativa m − n, garantendo l'equivarianza traslazionale della distribuzione di attenzione. RoPE è stato adottato in tutti i grandi modelli linguistici recenti (LLaMA, Gemma, Mistral) e viene impiegato nei Transformer per video world model di ultima generazione.

Per i modelli del mondo 3D spazio-temporali, la codifica posizionale si estende a tre dimensioni. V-JEPA adotta la fattorizzazione additiva:

$PE(t, h, w) = PE(t) + PE(h) + PE_w(w)$

dove t è l'indice temporale, h e w gli indici spaziali di riga e colonna. Questa fattorizzazione mantiene la separabilità spaziotemporale, riducendo il numero di parametri rispetto a una codifica congiunta sulle triple (t, h, w).

6. L'architettura Transformer completa: layer normalization, FFN e residual connections

Il blocco Transformer è composto da due sotto-blocchi, ciascuno avvolto da una connessione residua e normalizzazione di layer. Nella variante Pre-LayerNorm (adottata dalla maggior parte dei modelli moderni):

$X' = X + MHA(LayerNorm(X))$ $X'' = X' + FFN(LayerNorm(X'))$

dove LayerNorm normalizza le attivazioni per ogni posizione indipendentemente:

$$LayerNorm(x) = \gamma \odot (x - \mu(x)) / \sqrt{(\sigma^2(x) + \varepsilon)} + \beta$$

con $\mu(x) = (1/d) \Sigma\ x$ e $\sigma^2(x) = (1/d) \Sigma\ (x - \mu(x))^2$, e $\gamma, \beta \in \mathbb{R}^d$ sono parametri di scala e traslazione appresi. La normalizzazione Pre-LN, applicata all'input di ciascun sotto-blocco prima dell'operazione principale, produce gradienti più stabili rispetto alla Post-LN originale di Vaswani e collaboratori, ed è preferita per il training di modelli profondi senza warm-up del learning rate.

La rete feed-forward (FFN) applica una trasformazione posizione per posizione con proiezione a dimensione intermedia $d > d$ (tipicamente $d = 4d$):

$$FFN(x) = W_2\, \sigma(W_1\, x + b_1) + b_2$$

dove $W_1 \in \mathbb{R}^{d \times d}$, $W_2 \in \mathbb{R}^{d \times d}$, e σ è una non-linearità — tipicamente GELU (Gaussian Error Linear Unit):

$$GELU(x) = x \cdot \Phi(x) = x \cdot (1 + erf(x/\sqrt{2})) / 2$$

dove Φ è la CDF della gaussiana standard. La scelta di GELU rispetto a ReLU è motivata dalla differenziabilità ovunque e dalla soppressione morbida delle attivazioni vicine a zero, che produce gradienti più stabili. In molti Transformer moderni, σ viene sostituita con la variante SwiGLU:

$$SwiGLU(x, W_1, W_2, W_3) = (x\, W_1 \odot \sigma(x\, W_3))\, W_2$$

dove σ è la funzione sigmoide e $\odot$ il prodotto elemento per elemento. SwiGLU produce migliori performance empiriche a parità di parametri, al costo di un'architettura leggermente più complessa.

Le connessioni residue $X + f(X)$ hanno un effetto fondamentale sulla propagazione del gradiente: il gradiente della loss rispetto agli input profondi può raggiungere i

primi layer attraverso il percorso diretto della somma residua, evitando la catena di moltiplicazioni delle jacobiane dei sotto-blocchi. Formalmente, per un Transformer con L layer, il gradiente rispetto all'input del primo layer è:

$\partial \mathcal{L} / \partial X_0 = \partial \mathcal{L} / \partial X \cdot \Pi_{l=0}^{L-1} (I + J)$

dove J è la jacobiana del sotto-blocco l. La matrice identità I nel prodotto garantisce che il gradiente possa fluire senza attenuazione anche quando $\|J\| << 1$.

7. Flash-attention: algoritmo IO-aware per l'efficienza su hardware moderno

Il collo di bottiglia computazionale dell'attenzione scalata non è il numero di operazioni floating-point, ma il traffico di memoria (*memory bandwidth*) tra la GPU memory (DRAM, velocità ~2 TB/s) e i registri on-chip (SRAM, velocità ~19 TB/s). L'algoritmo standard di attenzione richiede di materializzare la matrice dei punteggi $S = QK^T/\sqrt{d} \in \mathbb{R}^{n \times n}$ e la matrice dei pesi $A = \text{softmax}(S) \in \mathbb{R}^{n \times n}$ in DRAM per permettere poi il calcolo di AV. Per $n = 4096$ e $d = 64$, la matrice S occupa $4096^2 \times 4$ byte ≈ 64 MB — largamente eccedente la capacità della SRAM (tipicamente 20 MB su una A100), e quindi il calcolo richiede molteplici trasferimenti tra DRAM e chip.

Dao e collaboratori (2022) introducono FlashAttention, un algoritmo che calcola l'output dell'attenzione senza mai materializzare la matrice n×n in DRAM, attraverso tiling e accumulazione incrementale sul chip. L'idea fondamentale sfrutta la struttura decomponibile della softmax. Per calcolare la riga i dell'output, è necessario:

$\tilde{y}_i = \Sigma \ \text{softmax}(e_i\square) \ v = \Sigma \ [\exp(e_i\square) / \Sigma \ \exp(e_i\square)] \ v$

Questo può essere scritto come:

$\tilde{y}_i = (\Sigma \ \exp(e_i\square) \ v) / (\Sigma \ \exp(e_i\square)) = N_i / D_i$

dove $N_i = \Sigma \ \exp(e_i\square) \ v$ e $D_i = \Sigma \ \exp(e_i\square)$ sono il numeratore e il denominatore della softmax normalizzata. Sia N e D accumulati incrementalmente su blocchi di colonne (tile) di K e V:

Per blocco b = 1, …, B (dove ogni blocco contiene B colonne): $S_b = Q K_o^T[b] / \sqrt{d}$ (punteggi per le colonne del blocco b) m_b = max(m_{b-1}, rowmax(S_b)) (running maximum per stabilità) N_b = exp(m_{b-1} - m_b) N_{b-1} + exp(S_b - m_b) V[b] (numeratore aggiornato) D_b = exp(m_{b-1} - m_b) D_{b-1} + rowsum(exp(S_b - m_b)) (denominatore aggiornato)

$\tilde{y}_i = N_B / D_B$

La divisione per exp(m_b − m_{b-1}) nella formula di aggiornamento è la tecnica del *running maximum* per la stabilità numerica della softmax: sottrarre il massimo corrente prima di esponenziale non cambia il risultato finale ma previene overflow.

L'efficienza di FlashAttention deriva dal fatto che le tile di Q, K, V sono caricate una volta in SRAM ed i calcoli sono completati interamente on-chip, eliminando i trasferimenti intermedi in DRAM. La complessità I/O di FlashAttention è $O(n^2 d / M)$ dove M è la capacità della SRAM, contro $O(n^2)$ dell'attenzione standard — un miglioramento di un fattore $d/M \approx 64/20 \approx 3\times$ per i parametri tipici delle GPU correnti. In pratica, FlashAttention produce uno speedup di 2–4× e riduce l'utilizzo della memoria da $O(n^2)$ a $O(n)$, rendendo praticabile l'addestramento con sequenze di decine di migliaia di token.

FlashAttention-2 (Dao 2023) perfeziona l'algoritmo riducendo il numero di operazioni non-matmul (che sono computazionalmente meno efficienti delle GEMM sulle GPU) e incrementando il parallelismo tra threadblock, raggiungendo fino all'80% della throughput teorica di picco dell'A100.

FlashAttention-3 (Shah e collaboratori 2024) sfrutta le capacità di asincronia dell'architettura Hopper (H100) attraverso la pipeline di warp specialization: il warpgroup di produzione esegue WGMMA (WarpGroup Matrix Multiply-Accumulate) per le operazioni di matmul, mentre il warpgroup di consumo esegue i trasferimenti di memoria TMA (Tensor Memory Accelerator) in parallelo, eliminando la latenza di accesso alla DRAM per blocchi di dimensioni opportune.

8. Causal masking e architetture autoreggressive nei modelli del mondo

L'attenzione standard permette a ogni posizione di "vedere" tutte le altre posizioni della sequenza, sia precedenti che successive. Questa configurazione — denominata attenzione bidirezionale (*bidirectional attention*) — è adatta per task di comprensione in cui l'intera sequenza è disponibile (classificazione, codifica). Per la generazione autoregressiva e per i modelli del mondo che devono predire lo stato futuro condizionando solo sulla storia passata, è necessaria l'attenzione causale (*causal masking*).

Il causal masking modifica la matrice dei punteggi sostituendo con $-\infty$ le entry corrispondenti a posizioni future:

$\tilde{S}_{ij} = S_{ij}$ se $j \leq i$ $\tilde{S}_{ij} = -\infty$ se $j > i$

Dopo la softmax, le entry con punteggio $-\infty$ producono pesi di attenzione esattamente zero, impedendo il flusso di informazione dalle posizioni future a quelle passate. Questa scelta trasforma il Transformer in un modello autoregressivo: la distribuzione condizionale $p(x_t \mid x_1, \ldots, x_{t-1})$ è calcolata parallelamente per tutti i t durante il training (attraverso il causal masking), ma la generazione rimane sequenziale durante l'inferenza, poiché x_t dipende da $x_1, \ldots, x_{t-1}$.

Per i world model autoregressivi — come IRIS, Genie, e i video diffusion Transformer — il causal masking è implementato come una maschera triangolare inferiore applicata alla matrice dei punteggi prima della softmax:

Mask = tril($J_{x\square}$) (matrice con 1 nella parte triangolare inferiore inclusa la diagonale)

In implementazioni efficienti, FlashAttention supporta nativamente il causal masking senza costo aggiuntivo attraverso il kernel specializzato: i blocchi di tile del triangolo superiore vengono semplicemente saltati, riducendo il numero di FLOP di un fattore ≈ 2 rispetto all'attenzione bidirezionale per sequenze lunghe.

9. Architetture Transformer per sequenze multimodali nei world model

I world model contemporanei operano su sequenze multimodali — frame video, istruzioni testuali, azioni robotiche, reward scalari, token posizionali — che devono essere integrati in un'unica rappresentazione contestuale. Il Transformer, grazie alla sua architettura flessibile (qualsiasi sequenza di vettori di dimensione d può essere processata), si presta naturalmente a questa integrazione multimodale.

La sfida principale è la tokenizzazione eterogenea: i frame video vengono tokenizzati come patch spaziotemporali attraverso un VQVAE o un patchifier convoluzionale; il testo viene tokenizzato attraverso BPE (Byte-Pair Encoding) o SentencePiece; le azioni robotiche possono essere tokenizzate come vettori continui discretizzati (RT-2) o come vettori continui con proiezione lineare (OpenVLA). I token di natura diversa vengono proiettati nello stesso spazio di dimensione d attraverso proiezioni lineari modalità-specifiche, e poi concatenati in una sequenza unica su cui opera il Transformer.

La distribuzione dell'attenzione in questi modelli multimodali presenta strutture di cross-modalità interessanti. In RT-2 e in VLM robotici, l'analisi delle mappe di attenzione rivela che i token di testo (le istruzioni) ricevono elevata attenzione dai token visivi corrispondenti agli oggetti rilevanti per l'istruzione — un meccanismo di grounding visivo emergente. Nei modelli del mondo video-testo (MAGVIT-v2, VideoGPT), l'attenzione crociata tra token video e token di azione permette al Transformer di condizionare la generazione video sulla sequenza di azioni, producendo simulazioni fisicamente plausibili.

Una variante architetturale importante è il *cross-attention* Transformer, usato in architetture encoder-decoder: il decoder calcola l'attenzione non solo sui propri token passati (self-attention causale) ma anche sull'intera sequenza codificata dall'encoder attraverso un meccanismo di attenzione crociata:

$$\text{CrossAttn}(Q_{dec}, K_{enc}, V_{enc}) = \text{softmax}(Q_{dec} K_{enc}^{T} / \sqrt{d}) V_{enc}$$

dove Q_{dec} sono le query del decoder e K_{enc}, V_{enc} sono chiavi e valori dell'encoder. Questa struttura è adottata in DALL-E, Stable Diffusion (nel condizionamento sul testo attraverso cross-attention), e nei modelli del mondo condizionati su linguaggio. In Dreamer condizionato su istruzioni linguistiche, il cross-attention permette al modello di transizione dell'RSSM di "leggere" l'istruzione

corrente durante il rollout immaginario, modulando la dinamica simulata in funzione dell'obiettivo linguistico.

10. Transformer per world model visivi: ViT, BERT visivo e MAE

Il Vision Transformer (ViT, Dosovitskiy e collaboratori 2020) ha portato l'architettura Transformer alla visione attraverso la divisione dell'immagine in patch non sovrapposte di dimensione $p \times p$, proiettate linearmente in vettori d-dimensionali e processate da un Transformer standard. Data un'immagine $I \in \mathbb{R}^{HxWxc}$, il numero di patch è $N = HW/p^2$, ciascuna di dimensione $p^2 \times c$. La proiezione lineare è:

$z_i = x_i W_e + b_e, i = 1, \ldots, N$

dove $x_i \in \mathbb{R}^{p2xc}$ è la patch i "appiattita" e $W_e \in \mathbb{R}^{p2cxd}$ è la matrice di proiezione dell'embedding. Un token di classe [CLS] viene prepeso alla sequenza per task di classificazione. La sequenza $z_0, z_1, \ldots, z$ (dove z_0 è il token CLS con embedding iniziale appreso) viene processata da L blocchi Transformer, e la rappresentazione z_0^(L) dopo l'ultimo blocco viene usata per la classificazione.

La complessità computazionale del ViT è $O(N^2 d)$ per l'attenzione — quadratica nel numero di patch. Per immagini ad alta risoluzione ($H = W = 512$, $p = 16$, $N = 1024$), questo è gestibile; per sequenze video di lunghezza T ($N = 1024 \times T$), la complessità diventa $O(N^2T^2d)$ — rapidamente ingestibile per $T > 10$. Le soluzioni pratiche includono: attenzione spaziotemporale fattorizzata (separate spatial e temporal attention heads), attenzione su frame selezionati (sparse temporal attention), e compressione delle patch video attraverso token summarization.

L'I-JEPA, discussa estesamente in precedenza, sfrutta il ViT come backbone sia per l'encoder di contesto che per l'encoder di target (aggiornato con EMA). Il predittore h_φ è implementato come un piccolo Transformer (tipicamente 4–8 layer con heads 8) che riceve i token di contesto codificati e i token posizionali delle regioni target, producendo le previsioni delle rappresentazioni target. La scelta di un predittore Transformer (invece di un semplice MLP) è motivata dalla necessità di integrare informazioni non locali dal contesto — il predittore deve "ragionare" su quali informazioni del contesto sono rilevanti per predire ogni posizione target.

11. Efficient Transformers per sequenze video: Sparse Attention e SSM

La limitazione quadratica dei Transformer standard per sequenze video lunghe ha stimolato due linee di ricerca parallele: gli *efficient Transformers* che modificano il pattern di attenzione per ridurre la complessità, e i *State Space Models* (SSM) che sostituiscono l'attenzione con ricorrenze lineari di nuova concezione.

Tra gli efficient Transformer per video, Longformer (Beltagy e collaboratori 2020) e BigBird (Zaheer e collaboratori 2020) introducono pattern di attenzione sparsi: ogni token attende solo ai token nella sua finestra locale (attenzione locale di finestra w → complessità O(nw)), più un insieme di token globali che attendono all'intera sequenza. Per i world model video, l'attenzione locale corrisponde a dipendenze spaziotemporali a corto raggio (frame vicini, patch vicine), mentre i token globali codificano strutture globali della scena.

I modelli a spazio di stato (*State Space Models*, SSM) come S4 (Gu e collaboratori 2021), H3, e Mamba (Gu e Dao 2023) propongono un'alternativa radicalmente diversa all'attenzione: la ricorrenza lineare con kernel convoluzionale appreso. Un SSM discretizzato è:

h = A h -1 + B x (transizione di stato) y = C h + D x (osservazione)

dove A ∈ $\mathbb{R}^{NxN}$, B ∈ $\mathbb{R}^{Nx1}$, C ∈ $\mathbb{R}^{1xN}$, D ∈ $\mathbb{R}$ sono parametri. Per matrici A strutturate (diagonali complesse con autovalori di modulo < 1), la risposta impulsiva y = Σ ≤ C^{At-s} B x è una convoluzione causale con kernel exponenzialmente decadente. Durante il training, questa convoluzione può essere calcolata in O(n log n) usando la FFT; durante l'inferenza, la ricorrenza è O(1) per token — molto più efficiente dell'attenzione O(n) per la gestione del KV-cache.

Mamba introduce la *selective SSM*: i parametri B, C e lo step size Δ dipendono dall'input x , rendendo il modello adattivo al contenuto invece che puramente convoluzionale. Questa selettività permette a Mamba di catturare dipendenze a lungo raggio in modo selettivo, analogamente all'attenzione ma con complessità lineare. Le architetture ibride Mamba-Transformer, come Jamba, combinano blocchi di attenzione (che eccellono nel ragionamento context-dependent) con blocchi Mamba (che gestiscono efficientemente sequenze molto lunghe), e costituiscono una direzione promettente per i world model video su sequenze di minuti.

12. Scala e capacità emergenti nei Transformer per world model

La scala delle architetture Transformer nei world model visivi segue le stesse leggi di scala scoperte per i modelli linguistici, con differenze negli esponenti che riflettono la differente struttura statistica dell'informazione visiva. Per i modelli del mondo video, Henighan e collaboratori mostrano che la loss di predizione autoregressiva dei token VQVAE scala come:

L(C) ≈ L_{oo} + A_vid / C^{α_vid}

con α_vid ≈ 0.065 — un esponente inferiore all'analogo linguistico α_lang ≈ 0.050, suggerendo che la modellazione video è più difficile e richiede più compute per unità di riduzione della loss.

Oltre la quantità, la scala dei Transformer nei world model porta a capacità qualitative che risultano difficilmente prevedibili a partire dalla sola riduzione della loss. GENIE, addestrato su 30.000 ore di video di videogiochi, sviluppa la capacità di generare ambienti interattivi coerenti da un singolo frame iniziale — una proprietà che non emerge nei modelli più piccoli addestrati sullo stesso dataset. SORA, con miliardi di parametri addestrati su un corpus video non divulgato di scala enorme, mostra comprensione della fisica newtoniana (traiettorie paraboliche, deformazione di materiali elastici, riflessione sulla superficie dell'acqua) che emerge spontaneamente come sottoprodotto dell'addestramento a predire video. Queste emergenze qualitative non sono predicibili dalla loss di training — un fatto che motiva la necessità di benchmark di valutazione fisicamente strutturati, come proposto in PhyGenBench, che testano la comprensione causale invece della sola qualità percettiva dei video generati.

La convergenza verso architetture Transformer uniformi — che trattano frame, patch, azioni, reward e istruzioni linguistiche come sequenze di token — rappresenta un momento di notevole unificazione nell'architettura dei world model. La capacità di integrare modalità eterogenee in un'unica operazione di attenzione, di sfruttare il pretraining su dati visivi e linguistici a scala internet, e di scalare le prestazioni in modo prevedibile con il compute, ha reso il Transformer l'architettura di riferimento per i world model della nuova generazione — incluse le varianti gerarchiche della JEPA che LeCun delinea come il percorso verso l'intelligenza autonoma di tipo animale.

XIV. Apprendimento per rinforzo con modelli del mondo: algoritmi model-based, Dyna, MuZero e pianificazione MCTS

1. Il dualismo model-free e model-based: una distinzione fondativa

L'apprendimento per rinforzo può essere suddiviso in due tradizioni epistemologicamente distinte, la cui differenza non è meramente tecnica ma riflette concezioni diverse di come un agente debba organizzare la propria conoscenza del mondo. L'approccio model-free — esemplificato da Q-learning, SARSA, e le loro varianti profonde come DQN, PPO, SAC — apprende direttamente una funzione di valore o una politica a partire dall'esperienza, senza costruire alcuna rappresentazione esplicita della dinamica ambientale. L'approccio model-based, al contrario, dedica una parte delle proprie risorse computazionali e rappresentative alla costruzione e all'aggiornamento di un modello del mondo — una struttura che codifica la distribuzione degli stati futuri condizionata sulle azioni passate — e utilizza questa struttura per pianificare, prevedere e accelerare l'acquisizione della politica.

La distinzione ha conseguenze dirette sull'efficienza del campionamento (*sample efficiency*). Un agente model-free deve raccogliere un numero elevato di interazioni reali con l'ambiente per convergere a una politica soddisfacente, perché ogni aggiornamento dei parametri sfrutta esclusivamente le transizioni effettivamente osservate. Un agente model-based può, in linea di principio, sfruttare ogni interazione reale in modo molto più intenso: il modello del mondo apprende da essa, e poi genera — a costo computazionale invece che sperimentale — un numero arbitrario di transizioni simulate su cui addestrare ulteriormente la politica. Questo meccanismo di amplificazione della conoscenza, proposto nella sua forma archetipica da Dyna, è il nucleo teorico dell'approccio model-based.

La tensione tra le due tradizioni non è priva di sfumature: il modello del mondo introduce inevitabilmente un errore di approssimazione che si propaga alla politica — il già discusso problema del compounding error. Un modello imperfetto può guidare la pianificazione verso regioni dello spazio degli stati dove il modello è sistematicamente errato (il cosiddetto *model exploitation* in opposizione al *model exploration*), producendo politiche che funzionano bene nell'ambiente simulato ma male nell'ambiente reale. L'equilibrio tra sfruttamento del modello e verifica empirica mediante interazione reale è la sfida fondamentale che i migliori algoritmi model-based — da Dyna a MuZero — affrontano con soluzioni diverse e complementari.

2. Il framework Dyna: amplificazione computazionale dell'esperienza

Richard Sutton propose l'architettura Dyna nel 1990 come soluzione elegante al problema dell'efficienza del campionamento, formalizzando l'idea che la pianificazione e l'apprendimento diretto dall'esperienza potessero procedere in parallelo, condividendo le stesse strutture rappresentative. La versione tabulare di Dyna-Q è descrivibile attraverso i tre componenti interagenti che la costituiscono.

Il primo componente è l'apprendimento diretto del modello: a ogni passo temporale in cui l'agente osserva la transizione (s, a, r_{+1}, s_{+1}), il modello deterministico $M(s, a) = (\hat{r}, \hat{s}')$ viene aggiornato per registrare la transizione osservata. In ambienti stocastici, il modello accumula le distribuzioni empiriche delle transizioni dallo stato s sotto l'azione a.

Il secondo componente è l'aggiornamento diretto dei valori dalla transizione reale, come in Q-learning standard:

$$Q(s, a) \leftarrow Q(s, a) + \alpha \left[r_{+1} + \gamma \max_{a'} Q(s_{+1}, a') - Q(s, a) \right]$$

Il terzo componente, distintivo di Dyna, è la pianificazione indiretta (*background planning*): per n iterazioni computazionali, un passo casuale precedentemente osservato $(\tilde{s}, \tilde{a})$ viene campionato dalla memoria, il modello predice la transizione $(\tilde{r}, \tilde{s}') = M(\tilde{s}, \tilde{a})$, e si esegue un aggiornamento Q-learning sulla transizione simulata:

$$Q(\tilde{s}, \tilde{a}) \leftarrow Q(\tilde{s}, \tilde{a}) + \alpha \left[\tilde{r} + \gamma \max_{a'} Q(\tilde{s}', a') - Q(\tilde{s}, \tilde{a}) \right]$$

Il parametro n — il numero di passi di pianificazione per ogni passo reale — controlla il trade-off tra costo computazionale e efficienza del campionamento. Per n = 0, Dyna-Q si riduce al Q-learning standard; per n grande, la politica converge molto più velocemente in termini di interazioni reali, al costo di più calcolo per ogni interazione. Sutton dimostrò empiricamente che su ambienti a labirinto, Dyna-Q con n = 50 raggiungeva la convergenza con circa un quindicesimo delle interazioni reali richieste dal Q-learning puro.

L'estensione profonda di Dyna sostituisce la tabella Q con una rete neurale Q_θ e il modello tabulare M con un modello parametrico neurale M_φ. L'architettura Dyna-Q profonda procede quindi come segue: le transizioni reali aggiornano sia M_φ (attraverso la loss di predizione del modello) sia Q_θ (attraverso la loss TD); le transizioni simulate da M_φ aggiornano ulteriormente Q_θ. Il problema critico nell'implementazione profonda è la propagazione degli errori: se M_φ produce transizioni sistematicamente distorte per alcune regioni dello spazio degli stati, le n iterazioni di pianificazione amplificano questo errore, potenzialmente destabilizzando il training di Q_θ. Le soluzioni pratiche includono l'uso di modelli ensemble per misurare l'incertezza epistemica e restringere il planning alle regioni a bassa incertezza, e la limitazione dell'orizzonte di rollout a $k \leq 3$–5 passi per contenere il compounding error.

3. Funzioni di valore e ottimizzazione della politica: formalismo rigoroso

Prima di procedere alle architetture più sofisticate, è utile fissare il formalismo delle funzioni di valore nei MDP, su cui si appoggiano tutti gli algoritmi model-based. Un Processo Decisionale di Markov (MDP) è la quintupla $(\mathcal{S}, \mathcal{A}, T, R, \gamma)$ con $T: \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ la funzione di transizione, $R: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ la funzione di ricompensa, e $\gamma \in [0, 1)$ il fattore di sconto. Il ritorno scontato a partire dal tempo t è il valore cumulato delle ricompense future:

$$G_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$$

La funzione di valore di stato $V^\pi(s)$ sotto la politica π è il ritorno atteso iniziando dallo stato s e seguendo π:

$$V^\pi(s) = \mathbb{E}_\pi [G_t \mid s_t = s] = \mathbb{E}_\pi \left[\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \mid s_t = s\right]$$

L'equazione di Bellman per V^π è la ricorrenza fondamentale che lega il valore in s al valore nei suoi successori:

$$V^\pi(s) = \sum_a \pi(a \mid s) \sum_{s'} T(s' \mid s,a) \left[R(s,a,s') + \gamma V^\pi(s')\right]$$

La funzione di azione-valore $Q^\pi(s, a)$ estende V^π alla coppia (stato, azione):

$$Q^\pi(s, a) = \sum_{s'} T(s' \mid s,a) \left[R(s,a,s') + \gamma \sum_{a'} \pi(a' \mid s') Q^\pi(s', a')\right]$$

La politica ottimale π^* massimizza $V^\pi(s)$ per ogni $s \in \mathcal{S}$ simultaneamente, e le corrispondenti funzioni ottimali $V^*(s) = \max_\pi V^\pi(s)$ e $Q^*(s,a) = \max_\pi Q^\pi(s,a)$ soddisfano l'equazione di Bellman ottimale:

$$Q^*(s, a) = \Sigma_{s'} T(s' \mid s,a) \left[R(s,a,s') + \gamma \max_{a'} Q^*(s', a')\right]$$

La politica ottimale si recupera come $\pi^*(s) = \text{argmax}_a Q^*(s, a)$. Il punto fondamentale che connette questo formalismo ai modelli del mondo è che le equazioni di Bellman richiedono esplicitamente la conoscenza della funzione di transizione T e della funzione di ricompensa R — precisamente le due componenti di un modello del mondo. Un agente model-based che disponga di una stima accurata $(\hat{T}, \hat{R})$ può computare, anche approssimativamente, le funzioni di valore ottimali attraverso la pianificazione, senza raccogliere nuova esperienza reale.

4. Model Predictive Control come pianificazione model-based a orizzonte finito

Il MPC, già discusso nella parte robotica di questa monografia, costituisce la strategia di pianificazione model-based più naturale in ambienti con dinamica continua. Il problema di ottimizzazione che il MPC risolve ad ogni passo temporale è:

$$\min_{a_t, \ldots, a_{t+H-1}} \Sigma_{\tau=t}^{t+H-1} c(\hat{s}_\tau, a_\tau) + V_f(\hat{s}_{t+H})$$

$$\text{soggetto a } \hat{s}_{\tau+1} = \hat{f}(\hat{s}_\tau, a_\tau), \ \tau = t, \ldots, t+H-1$$

dove $\hat{f}$ è il modello del mondo (funzione di transizione stimata), c la funzione di costo corrente, V_f il costo terminale, e H l'orizzonte di pianificazione. Nella variante *shooting*, le azioni $a_t, \ldots, a_{t+H-1}$ sono le variabili di ottimizzazione, e il gradiente del costo rispetto ad esse si ottiene retropropagando attraverso i passi del modello — il già descritto BPTT applicato al rollout.

Nella variante *shooting stocastico*, le azioni non vengono ottimizzate gradiente per gradiente ma campionate da una distribuzione, e la sequenza con il costo più basso viene selezionata. Il Cross-Entropy Method (CEM) — già introdotto nella sezione su PlaNet — è la realizzazione più comune di questo approccio: la distribuzione guida su sequenze di azioni è una gaussiana multivariata inizialmente isotropa, aggiornata iterativamente verso le sequenze a basso costo selezionate (elite set). Dopo J iterazioni di raffinamento, la media della distribuzione fornisce l'azione ottimale approssimata da applicare.

L'algoritmo MPPI (Model Predictive Path Integral, Williams e collaboratori 2017) offre un'alternativa basata sulla teoria del path integral del controllo stocastico. La formula di aggiornamento è:

$$a^* \approx \Sigma \ w \ a^{(k)} \ / \ \Sigma \ w \ , \text{con } w \ = \exp(-S(\tau)/\lambda)$$

dove $S(\tau) = \Sigma_{\tau} [c(\hat{s}_{\tau}^{(k)}, a\tau^{(k)}) + (\lambda/\varrho)\, a\tau^{(k)T} \Sigma^{-1} \varepsilon^{(k)}]$ è il costo perturbato della traiettoria k-esima, $\varepsilon^{(k)}$ è la perturbazione di rumore campionata, ϱ è la varianza del rumore, e λ è la temperatura. MPPI è derivabile dal principio del minimum free energy del controllo stocastico ottimo: la distribuzione ottimale sulle sequenze di azioni è la distribuzione di rumore pesata esponenzialmente dall'exponential-negativo del costo. La grandezza λ controlla il trade-off tra esplorazione (grande $\lambda \rightarrow$ distribuzione quasi-uniforme) e sfruttamento (piccolo $\lambda \rightarrow$ concentrazione sulle traiettorie a minor costo).

5. MCTS: pianificazione tramite ricerca ad albero con simulazioni

La Monte Carlo Tree Search (MCTS) è l'algoritmo di pianificazione che, combinato con reti neurali profonde per la valutazione degli stati, ha prodotto i sistemi di gioco più forti mai costruiti — da AlphaGo a AlphaZero a MuZero. L'idea fondamentale è di costruire incrementalmente un albero di ricerca nello spazio degli stati, usando simulazioni Monte Carlo per stimare il valore degli stati non ancora esplorati e concentrare la ricerca nelle regioni dell'albero più promettenti.

L'algoritmo MCTS si articola in quattro fasi ciclicamente ripetute:

Selezione: a partire dalla radice (stato corrente), si attraversa l'albero seguendo la politica di selezione UCB (Upper Confidence Bound) fino a raggiungere un nodo foglia non espanso. In ogni nodo s, l'azione a viene scelta massimizzando il punteggio UCB:

$$a^* = \text{argmax}_a \left[Q(s,a)/N(s,a) + c_puct\, P(s,a)\, \sqrt{(\Sigma_{a'} N(s,a'))} / (1 + N(s,a))\right]$$

dove $Q(s,a)$ è la somma dei valori delle simulazioni che hanno attraversato l'arco (s,a), $N(s,a)$ è il conteggio di visita dell'arco, $P(s,a)$ è la probabilità a priori dell'azione a nello stato s (fornita dalla rete di politica), e c_puct è una costante che controlla il bilanciamento tra sfruttamento (primo termine) ed esplorazione (secondo termine). Il termine UCB aumenta per azioni poco visitate ($N(s,a)$ piccolo) e per azioni con alta probabilità a priori, guidando la ricerca verso rami inesplorati ma promettenti.

Espansione: il nodo foglia selezionato viene espanso aggiungendo all'albero i suoi nodi figli (uno per azione), e la rete di valutazione fornisce una stima iniziale del valore $V(s)$ e della distribuzione di probabilità $P(s, \cdot)$ sulle azioni.

Simulazione (backup attraverso rollout o stima di valore): nel MCTS tradizionale, si esegue un rollout simulato seguendo una politica di rollout (tipicamente uniforme o leggermente informata) fino a uno stato terminale, e il risultato (vittoria/sconfitta/ritorno) viene registrato. Nei sistemi con reti neurali (AlphaGo, AlphaZero, MuZero), il rollout viene sostituito dalla stima di valore della rete: $V(s_leaf)$ fornisce direttamente la stima del ritorno atteso, eliminando la necessità di rollout lunghi.

Retroproiezione (backup): il valore V(s_leaf) viene propagato all'indietro lungo il percorso di selezione, aggiornando i contatori di visita e i valori cumulati per ogni nodo attraversato:

per ogni (s, a) nel percorso di selezione (in ordine inverso): N(s, a) ← N(s, a) + 1 W(s, a) ← W(s, a) + V(s_leaf) (somma dei valori) Q(s, a) ← W(s, a) / N(s, a) (valore medio)

Dopo N_sim simulazioni (tipicamente 50–800 a seconda del budget computazionale), la distribuzione delle visite ai figli della radice — π (a) = N(s , a) / Σ_{a'} N(s , a') — fornisce la politica di selezione dell'azione migliorata rispetto alla rete di politica originale. Questa distribuzione è il target di training per la rete di politica, in modo che l'agente impari a produrre distribuzioni di probabilità sempre più allineate con le politiche MCTS ottimizzate.

La convergenza del MCTS alla politica ottimale è garantita, nel limite del numero di simulazioni, per ambienti a due giocatori a somma zero con informazione perfetta (teorema di Kocsis e Szepesvári 2006 per UCT). Per MDPs a giocatore singolo e ambienti stocastici, la convergenza asintotica all'ottimo vale sotto condizioni simili di esplorazione sufficiente.

6. AlphaZero: apprendimento end-to-end con MCTS e reti profonde

AlphaZero (Silver e collaboratori 2018) è il sistema che ha dimostrato la potenza dell'unificazione tra reti neurali profonde e MCTS in un framework di apprendimento per rinforzo completamente self-play, senza alcuna conoscenza di dominio oltre alle regole del gioco. L'architettura di AlphaZero è una rete residuale profonda (ResNet con blocchi di tipo residuo — tipicamente 20 blocchi con 256 filtri per gli scacchi) che prende in input la rappresentazione dello stato del gioco e produce due output congiuntamente:

$$(P(s, \cdot), V(s)) = f_\theta(s)$$

dove $P(s, \cdot) \in \Delta(\mathcal{A})$ è la distribuzione di probabilità sulle mosse (prior per MCTS) e $V(s) \in [-1, 1]$ è la stima del valore dello stato (outcome atteso della partita).

Il training di AlphaZero è completamente self-play: a ogni iterazione, l'agente gioca contro se stesso usando MCTS per selezionare le mosse, raccoglie dati di training nella forma (s , π , z) dove π è la distribuzione MCTS e $z \in \{+1, -1\}$ è l'outcome finale della partita. La funzione di perdita è:

$$\mathcal{L}(\theta) = -(1/T) \Sigma \ [\pi^{T} \log P_\theta(s , \cdot) + (z - V_\theta(s))^2] + \lambda \|\theta\|^2$$

Il primo termine è la cross-entropy tra la distribuzione MCTS (target più accurata) e la rete di politica (modello corrente) — il MCTS "insegna" alla rete ad approssimare la politica di pianificazione. Il secondo termine è l'MSE tra la stima di valore della rete e

l'outcome reale della partita. Il terzo termine è la regolarizzazione L_2. Il learning loop è quindi:

1. Generare self-play data con MCTS guidato da f_θ
2. Aggiornare θ minimizzando $\mathcal{L}$ sui dati recenti
3. Ripetere

La convergenza di AlphaZero è notevole per la sua rapidità e la qualità del risultato: il sistema supera i campioni mondiali di scacchi, shogi e go dopo rispettivamente 4, 2 e 30 ore di training su una singola macchina TPU. La chiave del successo è la sinergia tra MCTS e rete neurale: il MCTS produce target di training più accurati dei target online del Q-learning puro (grazie alla ricerca nel futuro), mentre la rete neurale comprime la conoscenza acquisita in un formato che può essere usato efficientemente come prior nelle simulazioni MCTS.

7. MuZero: apprendimento del modello del mondo per la pianificazione

MuZero (Schrittwieser e collaboratori 2020) rappresenta il salto concettuale più significativo da AlphaZero nella teoria dei world model per la pianificazione: esso apprende non solo la politica e la funzione di valore, ma anche un modello del mondo latente — tutto ciò di cui ha bisogno per eseguire MCTS — senza richiedere la conoscenza esplicita delle regole del gioco o della struttura dell'ambiente.

L'architettura di MuZero si compone di tre funzioni neurali parametriche:

La **funzione di rappresentazione** h_θ: $\mathcal{O} \to \mathcal{S}$ mappa la storia delle osservazioni passate $(o_1, \ldots, o_t)$ in uno stato latente $s_t = h_\theta(o_1, \ldots, o_t) \in \mathbb{R}^d$.

La **funzione di dinamica** g_θ: $\mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathbb{R}$ mappa la coppia (stato latente, azione) nel successivo stato latente e nella ricompensa predetta:

$$(s_{t+1}, \hat{r}_{t+1}) = g_\theta(s_t, a_t)$$

La **funzione di predizione** f_θ: $\mathcal{S} \to \Delta(\mathcal{A}) \times \mathbb{R}$ produce la distribuzione di politica e la stima di valore dallo stato latente:

$$(p_t, v_t) = f_\theta(s_t)$$

La pianificazione MCTS in MuZero opera interamente nello spazio latente, senza accesso all'osservazione reale: dato lo stato radice $s_0 = h_\theta(o_1, \ldots, o_t)$, il MCTS simula K passi di dinamica latente g_θ per espandere l'albero, usando f_θ per valutare ogni nodo e fornire il prior di politica. Le simulazioni non richiedono mai di interagire con l'ambiente reale — sono esclusivamente computazioni sulle reti neurali g_θ e f_θ.

La funzione di perdita di MuZero addestra congiuntamente le tre componenti con target derivati sia dall'osservazione reale sia dalle distribuzioni MCTS:

$\mathcal{L}(\theta) = \Sigma_{t=0}^{T} \Sigma_{k=0}^{K} [\ell^r(u_{+\square}, \hat{r}_{\square+\square}) + \ell^v(z_{+\square}, v_{+\square}) + \ell^p(\pi_{+\square}, p_{+\square})]$

dove: $\ell^r(u, \hat{r}) = (u - \hat{r})^2$ è la MSE per la ricompensa (target u dalla ricompensa reale) $\ell^v(z, v) = (z - v)^2$ è la MSE per il valore (target z dall'n-step return MCTS) $\ell^p(\pi, p) = -\pi^T \log p$ è la cross-entropy per la politica (target π dalla distribuzione MCTS)

e l'indice k scorre i K passi di rollout nel modello latente. Il rollout multipassso (K > 1) nella loss è fondamentale: forza il modello di dinamica g_θ a produrre stati latenti accuratamente predittivi per almeno K passi nel futuro, invece di limitarsi al singolo passo successivo.

Una proprietà notevole di MuZero è che lo spazio latente s non è vincolato a ricostruire l'osservazione originale o : non vi è alcun termine di ricostruzione percettiva nella loss. Lo stato latente deve solo essere predittivo delle ricompense future e delle distribuzioni di politica MCTS — esattamente il principio della JEPA applicato alla pianificazione con rinforzo. Questo consente allo spazio latente di ignorare i dettagli percettivi irrilevanti per la pianificazione (texture, illuminazione) e di concentrare la propria capacità rappresentativa sulle strutture dinamicamente rilevanti — esattamente l'intuizione di LeCun sulla superiorità della predizione nello spazio latente.

8. MuZero Reanalyze e l'efficienza del campionamento

MuZero Reanalyze (Schrittwieser e collaboratori 2021) estende l'architettura per massimizzare l'efficienza del campionamento, affrontando un limite dell'algoritmo originale: le distribuzioni MCTS usate come target di training vengono calcolate con il modello alla versione $\tilde{\theta}$ precedente all'aggiornamento corrente, e invecchiano rapidamente man mano che il modello migliora. Reanalyze riesegue periodicamente il MCTS sui dati già raccolti usando il modello aggiornato:

per ogni traiettoria (o_1, …, o , a_1, …, a , r_1, …, r) nel replay buffer: $s_0 = h_\theta(o_1$, …, o _start) per t = t_start, …, t_end: eseguire MCTS con modello aggiornato f_θ, g_θ registrare (s , π , z) con nuovi target

In questo modo, ogni transizione nel replay buffer viene usata più volte, con target di policy sempre più accurati man mano che il modello migliora. Su Atari 100K (100.000 frame di interazione reale), MuZero Reanalyze raggiunge prestazioni paragonabili ad agenti model-free addestrati su 10–20 volte più frame — una dimostrazione convincente dell'efficienza del campionamento dell'approccio model-based con MCTS.

Stochastic MuZero (Antonoglou e collaboratori 2022) estende il framework ad ambienti con transizioni stocastiche, introducendo un modello di dinamica stocastico che produce non uno stato deterministico ma una distribuzione di afterstates:

$z_{t+1} \sim q_\theta(z_{t+1} \mid s_t, a_t)$ $s_{t+1} = \tilde{h}_\theta(s_t, a_t, z_{t+1})$

dove $z_{t+1} \in \{1, \ldots, C\}$ è una variabile categoriale che parametrizza la transizione, e $\tilde{h}_\theta$ combina lo stato corrente, l'azione, e il campione della distribuzione per produrre lo stato successivo. La pianificazione MCTS avviene in un albero di ricerca che include nodi chance (i campioni z_{t+1}) alternati ai nodi decisione (le azioni), analogamente all'expectimax search per i giochi stocastici.

9. Efficienza computazionale del MCTS: parallellismo e ottimizzazione della ricerca

L'implementazione ad alte prestazioni del MCTS per MuZero richiede un'attenzione specifica all'efficienza computazionale, in quanto il training richiede miliardi di simulazioni nel modello latente. Due strategie principali consentono la scalatura.

Il **batch MCTS** raggruppa le richieste di valutazione della rete f_θ da simulazioni diverse in un singolo batch, sfruttando la throughput di GPU/TPU per la valutazione di molti stati in parallelo. Invece di eseguire le N_{sim} simulazioni sequenzialmente (aspettando il completamento di ognuna prima di iniziare la successiva), il batch MCTS mantiene un pool di simulazioni attive, avanzandole tutte fino al prossimo nodo foglia che richiede valutazione, poi eseguendo la valutazione in batch, poi ricominciando. Questo produce uno speedup proporzionale alla dimensione del batch di valutazione — tipicamente 8–64 × rispetto all'esecuzione sequenziale.

Il **virtual loss** è una tecnica per rendere il MCTS compatibile con il parallellismo: quando una simulazione seleziona l'arco (s, a), aggiunge artificialmente un contributo negativo al valore aggregato W(s, a) (il "virtual loss") che scoraggia le simulazioni parallele dal selezionare lo stesso arco immediatamente. Questo rende le simulazioni parallele più esplorative, riducendo la correlazione tra i loro percorsi nell'albero. Formalmente, se n_{vl} simulazioni parallele stanno attraversando l'arco (s, a), il valore usato nella selezione UCB viene abbassato di $n_{vl} \cdot \beta$ per un parametro di penalizzazione β.

L'**alphazero MCTS parallelism** (DeepMind) opera con 64 GPU di valutazione che servono simultaneamente 1024 attori (processi di simulazione indipendenti), producendo dati di self-play ad alta velocità per alimentare il training della rete. Il throughput del sistema è dell'ordine di 300.000 passi di simulazione al secondo per gli scacchi, sufficienti per generare la quantità di dati necessaria al training di un modello di alta qualità in 12–24 ore.

10. EfficientZero: trasferimento del paradigma MuZero ad ambienti visivi ad alta dimensione

EfficientZero (Ye e collaboratori 2021) adatta l'architettura MuZero agli ambienti Atari con osservazioni ad alta dimensione (immagini 84×84 con stack di 4 frame),

introducendo tre modifiche architetturali fondamentali per migliorare l'efficienza del campionamento.

La prima modifica è la **self-supervised consistency loss** tra la rappresentazione h_θ(o ₊□) dell'osservazione reale k passi nel futuro e la rappresentazione predetta dal rollout del modello latente g_θ^k(s):

ℒ_cons = Σ D(sg(h_θ(o ₊□)), g_θ^k(s))

dove sg(·) è lo stop-gradient applicato alla rappresentazione reale (per prevenire il collasso), e D è la distanza coseno. Questa loss aggiuntiva forza la coerenza tra le traiettorie nel spazio latente e le osservazioni reali, ancorando lo stato latente a caratteristiche percettive concrete invece di permettergli di derivare verso rappresentazioni astratte non ancorate.

La seconda modifica è l'**end-to-end learning del valore dell'n-step return nel modello**: i target di valore z per il training sono calcolati come n-step returns nel modello latente invece dell'outcome della partita completa, accelerando il segnale di apprendimento per stati distanti dalla fine della traiettoria.

La terza modifica è la **correzione della distribuzione** (*off-policy correction*): poiché le traiettorie nel replay buffer sono state raccolte con versioni precedenti del modello, EfficientZero corregge il bias off-policy usando importance weights basati sul rapporto tra la politica corrente e quella di raccolta.

Il risultato empirico è notevole: EfficientZero supera le prestazioni umane mediane su Atari 100K usando 100.000 frame — 400 volte meno dati rispetto all'efficiente agente Rainbow, e 200 volte meno rispetto a DreamerV2. Questo posiziona EfficientZero come l'algoritmo più efficiente nel campionamento per ambienti visivi discreti, e la sua architettura è oggi un punto di riferimento per lo sviluppo di world model per il gioco e la simulazione.

11. Integrazione tra MCTS, world model neurali e apprendimento continuo

La connessione tra MCTS e i world model del tipo Dreamer/RSSM è naturale ma non immediata: il MCTS opera su alberi di stati discreti con azioni discrete, mentre l'RSSM modella spazi di stato e azione continui. La loro integrazione richiede alcune astrazioni.

Muzero può essere visto come un caso specifico del framework RSSM in cui il modello di transizione è deterministico (k = 1 campione) e il rollout ha orizzonte K finito (invece dell'orizzonte infinito del ritorno scontato). La connessione formale si stabilisce considerando lo stato latente s ∈ ℝ^{d} di MuZero come equivalente al vettore (h , z) dell'RSSM, e la funzione g_θ come l'RSSM deterministico con z campionato dal prior invece che dalla posteriore.

Il Sampled MuZero (Hubert e collaboratori 2021) estende MuZero agli spazi di azioni continui, necessari per le applicazioni robotiche. L'idea è di campionare M azioni candidate a ~ P(s, ·) (da una distribuzione parametrica come una gaussiana o una normalizzazione flow) e usare solo queste M azioni come possibili espansioni dell'albero di ricerca. La distribuzione di politica P(s, ·) è una distribuzione continua parametrizzata da una rete, e il MCTS opera su questo insieme finito di azioni campionate per ogni nodo. I punteggi UCB sono adattati per l'azione campionata:

a* = argmax_{a } [Q(s, a)/N(s, a) + c_puct P(s, a) √(Σ_{a'} N(s, a')) / (1 + N(s, a))]

La coerenza del metodo richiede che M sia sufficientemente grande da approssimare adeguatamente la distribuzione ottimale sullo spazio continuo — tipicamente M = 8–16 azioni campionate per nodo sono sufficienti per spazi di azione di dimensione ≤ 10.

Per l'apprendimento continuo, l'integrazione più sofisticata è quella di Hafner (2024) con DreamerV3, che utilizza rollout dell'RSSM come sostituto del MCTS: la politica si addestra su traiettorie immaginarie di lunghezza H nel modello dell'RSSM, ottimizzando il λ-ritorno su queste traiettorie invece di usare la ricerca in albero. Il vantaggio è la differenziabilità end-to-end (il gradiente fluisce attraverso i passi del rollout imaginario) e la naturalità con lo spazio di azione continuo; lo svantaggio è la minore qualità della pianificazione rispetto al MCTS per decisioni richiedenti ragionamento combinatorio profondo.

12. Verso architetture unificate: MCTS, world model e fondamenti teorici della generalizzazione

Il campo dell'apprendimento per rinforzo con modelli del mondo è oggi attraversato da una tensione fruttuosa tra due visioni della pianificazione. Da un lato, la tradizione AlphaZero-MuZero sostiene che la ricerca ad albero — un processo computazionale esplicito di lookahead — è indispensabile per il ragionamento strategico a lungo orizzonte, e che nessuna rete neurale può approssimarlo in modo soddisfacente senza eseguire questa ricerca in modo esplicito. Dall'altro, la tradizione Dreamer-RSSM sostiene che l'ottimizzazione della politica attraverso gradienti analitici sul rollout immaginario è sufficiente, e che la ricerca ad albero è un onere computazionale giustificato solo per ambienti con struttura combinatoria discreta molto profonda.

I risultati empirici suggeriscono che entrambe le posizioni catturano verità parziali. MuZero domina su giochi come scacchi e go, dove l'orizzonte strategico è molto lungo e la struttura dell'albero è finemente ramificata. Dreamer domina su task di controllo continuo come il controllo del robot di manipolazione o la navigazione, dove il rollout differenziabile è naturalmente adattato allo spazio di azione. L'architettura che emergerà come dominante nei prossimi anni sarà probabilmente ibrida: un world model latente del tipo RSSM/JEPA che fornisce il modello di transizione per il MCTS, con il MCTS usato solo per le decisioni che richiedono

lookahead profondo, e il rollout differenziabile usato per l'ottimizzazione della politica a lungo termine.

La connessione teorica più profonda tra MCTS e teoria dell'apprendimento è quella con i bound di regret. L'algoritmo UCT (UCB applied to Trees, Kocsis e Szepesvári 2006) soddisfa il bound:

$$\mathbb{E}[\text{regret dopo } n \text{ simulazioni}] = O(n^{1/2} \log n)$$

che converge all'azione ottimale con un tasso di $O(\log n / n)$ per ogni azione subottimale. Questo bound, valido per MDP stazionari, garantisce la convergenza del MCTS all'ottimo con sufficiente budget computazionale. Nei world model con dinamiche non stazionarie (ambienti che cambiano durante il training, distributional shift), il bound si degrada, e la stabilità della pianificazione dipende criticamente dall'accuratezza del modello latente nelle regioni esplorate dall'albero di ricerca.

La questione aperta più rilevante è quella della generalizzazione composizionale della pianificazione: mentre il MCTS converge all'ottimo per ogni MDP stazionario individuale, non vi è garanzia che la struttura dell'albero di ricerca imparata su un insieme di MDP di training si trasferisca efficientemente a MDP di test con struttura diversa. La connessione con la teoria causale — sviluppata nel capitolo precedente — suggerisce che solo i modelli del mondo che apprendono strutture causali invarianti tra task potranno generalizzare la pianificazione a scenari non visti, combinando in modo compositivo le strutture causali apprese separatamente in un piano valido per il nuovo task. Questa integrazione tra world model causali e pianificazione MCTS rappresenta oggi una delle frontiere più attive e promettenti dell'intera disciplina.

XV. Modelli di linguaggio come modelli del mondo impliciti: dalla comprensione del testo alla simulazione del mondo fisico e sociale

1. Il linguaggio come compressione della struttura del mondo

L'idea che un modello addestrato a predire sequenze di testo possa sviluppare una comprensione del mondo fisico, sociale e causale non era affatto ovvia a priori, e rimase controversa per lungo tempo anche dopo che le evidenze empiriche cominciarono ad accumularsi. Il linguaggio, nella tradizione simbolica inaugurata da Frege e sviluppata dalla filosofia analitica del Novecento, è un sistema di rappresentazione convenzionale — i suoi simboli acquisiscono significato attraverso la convenzione e l'uso, non attraverso una relazione causale diretta con le entità che denotano. Come potrebbe allora una rete neurale che impara a predire il prossimo token in una sequenza testuale acquisire conoscenza del mondo fisico che quella sequenza descrive?

La risposta più soddisfacente a questa domanda si trova nella prospettiva informazionale. Il testo prodotto dagli esseri umani è il prodotto di processi cognitivi, percettivi e sociali profondamente radicati nell'interazione con il mondo fisico. Un essere umano che scrive "la palla rotolò giù per le scale e si fermò sul pianerottolo" sta codificando nella sequenza di simboli una quantità considerevole di conoscenza fisica implicita: la direzione del moto gravitazionale, la proprietà degli oggetti sferici di rotolare, la discontinuità geometrica delle scale, la decelerazione per attrito. Un modello che apprende a predire questa sequenza — e le centinaia di miliardi di sequenze simili presenti nel corpus di pre training — è costretto, pena l'impossibilità di ridurre la loss di predizione, a sviluppare rappresentazioni interne che catturino almeno una parte di questa struttura fisica implicita.

Mitchell e collaboratori (2022) hanno formalizzato questa intuizione attraverso la nozione di *world model implicito*: un modello di linguaggio apprende una rappresentazione compressa delle distribuzioni condizionali p(token successivo | storia), e questa distribuzione può essere fattorizzata attraverso uno stato interno s che cattura le caratteristiche semanticamente e fisicamente rilevanti della situazione descritta nel testo. Lo stato interno non è direttamente osservabile — è una struttura emergente delle rappresentazioni del Transformer — ma può essere sondato attraverso sonde lineari (*linear probing*) addestrate a prevedere proprietà fisiche del mondo descritto dal testo a partire dalle attivazioni intermedie del modello.

Li e collaboratori (2023) hanno dimostrato in modo convincente l'esistenza di rappresentazioni interne strutturate nei modelli di linguaggio attraverso esperimenti sul gioco di Othello. Un Transformer addestrato a predire sequenze di mosse legali di Othello — senza alcuna supervisione esplicita sulla struttura del tabellone — sviluppa nelle sue rappresentazioni interne una rappresentazione implicita e accurata dello stato del tabellone a ogni passo della partita. Una sonda lineare addestrata sulle attivazioni del Transformer predice lo stato di ogni cella del tabellone con accuratezza superiore al 99% — indistinguibile da una rappresentazione esplicita dello stato. Questo risultato, sebbene relativo a un ambiente semplice e strutturato, fornisce evidenza diretta che i Transformer apprendono modelli del mondo impliciti dallo stream di token, non semplici pattern statistici superficiali.

2. La struttura del pre training come apprendimento di un prior sul mondo

Dal punto di vista dei modelli del mondo, il pre training di un LLM su un corpus di testo può essere interpretato come l'apprendimento di un prior probabilistico estremamente ricco sulla struttura del mondo. Il corpus di pre training contiene descrizioni di eventi fisici, relazioni causali, stati affettivi degli agenti, strutture sociali, regole giuridiche, procedure tecniche — in sostanza, la sedimentazione linguistica di secoli di conoscenza umana sul funzionamento del mondo.

La distribuzione appressa dal modello di linguaggio $p_\theta(x \mid x_1, \ldots, x_{-1})$ può essere decomposta, seguendo la prospettiva bayesiana, come:

$$p_\theta(x \mid x_1, \ldots, x_{-1}) = \int p(x \mid s)\, p(s \mid x_1, \ldots, x_{-1})\, ds$$

dove s è lo stato del mondo implicito a cui si riferisce la sequenza testuale, $p(x \mid s)$ è il modello di emissione (come si descrive linguisticamente lo stato s), e $p(s \mid x_1, \ldots, x_{-1})$ è la distribuzione di credenza aggiornata sullo stato del mondo dato il testo precedente. In questa prospettiva, il processo autoregressivo di predizione del token implementa implicitamente un filtraggio bayesiano dello stato del mondo attraverso la sequenza testuale — esattamente la struttura del POMDP che abbiamo analizzato nel capitolo dedicato alla pianificazione sotto incertezza.

Questa lettura ha implicazioni concrete. Le allucinazioni dei modelli di linguaggio — la generazione di testo plausibile ma fattualmente errato — corrispondono a situazioni in cui la distribuzione implicita $p(s \mid x_1, \ldots, x_{-1})$ assegna alta probabilità a stati del mondo scorretti, perché la struttura del corpus di pre training ha associato pattern linguistici simili a quelli della query con informazioni false o fuorvianti. La calibrazione del modello — la corrispondenza tra la confidenza espressa e l'accuratezza fattuale — è la manifestazione linguistica della qualità del prior sul mondo acquisito durante il pre training.

Formalmente, un modello di linguaggio ben calibrato soddisfa:

$$p(\text{affermazione vera} \mid \text{il modello afferma con confidenza } p) \approx p$$

La calibrazione può essere misurata attraverso l'Expected Calibration Error (ECE):

$$ECE = \Sigma\ (|B| / n)\ |acc(B) - conf(B)|$$

dove i B sono bin di confidenza, $acc(B)$ è l'accuratezza media nel bin, $conf(B)$ è la confidenza media nel bin, e n è il numero totale di esempi. I modelli di linguaggio di grandi dimensioni mostrano sistematicamente overconfidence per affermazioni fattualmente errate — una forma di miscalibrazione che può essere corretta attraverso tecniche di scaling della temperatura, RLHF, o retrieval-augmented generation.

3. Chain-of-Thought come esternalizzazione della pianificazione latente

Il *chain-of-thought prompting* (Wei e collaboratori 2022) è la tecnica che ha rivelato in modo più chiaro le capacità di ragionamento dei grandi modelli di linguaggio come world model impliciti. L'idea è semplice: invece di chiedere al modello di produrre direttamente la risposta finale, si chiede di esplicitare i passi di ragionamento intermedi. La risposta si sviluppa come una sequenza di inferenze intermedie — il cosiddetto "pensiero a catena" — prima di raggiungere la conclusione.

Il fondamento teorico del chain-of-thought può essere espresso in termini di complessità descrittiva. Consideriamo un problema che richiede k passi di inferenza per essere risolto. Un modello che produce la risposta in un singolo passo deve comprimere questi k passi nella propagazione in avanti del singolo token di risposta

— una computazione di profondità fissa (L layer, d dimensioni) che può implementare al più O(L) passi di reasoning sequenziale se il reasoning richiede propagazione sequenziale di informazione. Al contrario, il chain-of-thought esternalizza il reasoning nel contesto testuale: ogni passo intermedio è un token che il modello produce e poi "legge" come parte del proprio contesto, permettendo a ogni passo successivo di accedere al risultato del precedente. La lunghezza della catena di reasoning diventa quindi il nuovo asse di scalatura computazionale — indipendente e aggiuntiva rispetto alla profondità e alla larghezza della rete.

Formalmente, sia P un problema con soluzione S che richiede k passi intermedi $c_1, \ldots, c$. La probabilità del modello standard di produrre S è:

$$p(S \mid P) = \Sigma_{c_1,\ldots,c} \; p(S \mid c, P)\, p(c \mid c_{-1}, P) \cdots p(c_1 \mid P)$$

che il modello approssima in modo compresso come $p(S \mid P) \approx f_\theta(S, P)$. Il chain-of-thought esplicita la catena di Markov latente:

$$p(S, c_1, \ldots, c \mid P) = p(S \mid c, P)\, \Pi_i \, p(c_i \mid c_{i-1}, P)$$

e il modello produce questa sequenza autoregressivamente: genera c_1, poi c_2 condizionato su $P \cup \{c_1\}$, e così via. Ogni c_i è un "stato del mondo intermedio" nella computazione — il risultato di un'operazione sul modello implicito del problema.

L'evidenza empirica mostra che per problemi richiedenti ragionamento matematico, ragionamento simbolico e ragionamento fisico, il chain-of-thought aumenta l'accuratezza dei modelli di 10–40 punti percentuali rispetto al prompting diretto. Questa differenza è monotona crescente con la dimensione del modello — i modelli più piccoli non beneficiano del chain-of-thought, mentre i modelli sopra un certo threshold di scala (circa 100B parametri nella misurazione originale di Wei e collaboratori) mostrano il beneficio maggiore. Questa emergenza scalare del reasoning è coerente con la tesi che il chain-of-thought sblocchi capacità latenti dei modelli di linguaggio come simulatori del mondo, capacità che esistono nelle rappresentazioni interne ma non sono esprimibili direttamente nel singolo token di risposta.

4. Ragionamento fisico nei modelli di linguaggio: evidenze e limiti

Le capacità dei modelli di linguaggio nel ragionamento fisico — cioè nel prevedere correttamente le conseguenze di azioni fisiche in scenari descritti in linguaggio naturale — sono state studiate sistematicamente attraverso benchmark come BIG-Bench Physical Intuition, PhysicsBenchmark e PIQA. I risultati mostrano un quadro sfumato di capacità e limitazioni.

GPT-4 e modelli comparabili risolvono con alta accuratezza problemi di fisica qualitativa che richiedono la conoscenza di leggi fisiche elementari: la caduta degli oggetti ("quale oggetto tocca terra per primo, lasciato cadere da pari altezza: una palla di piombo o una piuma nel vuoto?"), le proprietà termiche ("perché si usa ghiaccio per

raffreddare le bevande e non acqua fredda?"), i principi meccanici di base. Tuttavia, per problemi che richiedono ragionamento quantitativo preciso o la composizione di più leggi fisiche in configurazioni inusuali — mai descritte in modo esplicito nel corpus di pre training — le prestazioni degradano significativamente e le allucinazioni fisiche diventano frequenti.

La spiegazione più coerente con i dati è che i modelli di linguaggio apprendono, attraverso il pretraining, una forma di *intuitiva fisica* statistica: distribuzioni di probabilità sui comportamenti fisici attesi in situazioni descritte linguisticamente, apprese per imitation delle descrizioni fisiche nel corpus. Questa fisica intuitiva è accurata per le configurazioni rappresentate con sufficiente densità nel corpus (situazioni fisiche quotidiane, esperimenti da manuale), ma fallisce per le configurazioni composizionalmente nuove (combinazioni di principi fisici mai associate nello stesso testo).

Il confronto con la fisica intuitiva umana è istruttivo. Gli esseri umani commettono sistematicamente errori in certi domini di fisica intuitiva — la traiettoria di oggetti lanciati, il comportamento di sistemi a rotazione — ma mostrano una generalizzazione compositiva molto più robusta dei modelli di linguaggio per situazioni nuove. La differenza qualitativa è attribuita al fatto che la fisica intuitiva umana è fondata sull'esperienza sensorimotoria diretta — l'incarnazione discussa nel capitolo sulla robotica — mentre la fisica intuitiva dei modelli di linguaggio è puramente statistico-culturale. Questa distinzione motiva la proposta di connettere i modelli di linguaggio con sensori e attuatori fisici (come in RT-2 e OpenVLA) per fondare la comprensione del linguaggio nell'esperienza embodied.

5. Modelli di linguaggio e strutture sociali: la simulazione di agenti

Una dimensione della simulazione del mondo che i modelli di linguaggio sviluppano con qualità sorprendente è quella sociale e psicologica: la capacità di modellare gli stati mentali, le credenze, le preferenze e i comportamenti di agenti umani. Questa capacità — denominata *Teoria della Mente* computazionale (*Theory of Mind*, ToM) nella letteratura psicologica — è il fondamento del ragionamento sociale e della cooperazione negli esseri umani.

I benchmark classici di ToM per gli esseri umani, come il test della falsa credenza (*false belief task*) di Wimmer e Perner, testano la capacità di un individuo di attribuire ad un altro individuo una credenza diversa dalla propria. Nel test di Sally-Anne, un soggetto osserva che Sally mette una biglia nella sua cesta e poi lascia la stanza; durante la sua assenza, Anne sposta la biglia nella sua scatola. Quando Sally rientra, il soggetto viene chiesto: "dove cerca Sally la sua biglia?". La risposta corretta è "nella cesta" — dove Sally crede sia — non "nella scatola" — dove realmente si trova.

Kosinski (2023) ha testato GPT-4 e modelli comparabili su varianti di questi task, rilevando prestazioni superiori all'80% di accuratezza — comparabili a bambini di 7–8

anni. Successivi lavori (Ullman 2023, Shapira e collaboratori 2023) hanno mostrato che le prestazioni degradano significativamente su varianti minimali del task che cambiano dettagli superficiali senza modificare la struttura logica — suggerendo che i modelli risolvono i task di ToM attraverso pattern superficiali associati ai test nel corpus di pretraining piuttosto che attraverso genuina comprensione della struttura delle credenze.

Questa controversia — che rispecchia la più ampia discussione sull'emergenza di capacità genuine versus l'apprendimento di shortcuts statistiche — è aperta ed è metodologicamente intricata. Park e collaboratori (2023) hanno proposto un approccio diverso all'uso dei modelli di linguaggio come simulatori di agenti sociali: in "Generative Agents", i modelli di linguaggio simulano una comunità di agenti con memorie, caratteri e relazioni, producendo comportamenti sociali emergenti (formazione di amicizie, organizzazione di eventi, diffusione di informazioni) che mostrano una coerenza interna notevole su scale di simulazione di giorni. In questo contesto, il modello di linguaggio non è testato per la sua comprensione della ToM, ma usato come motore di simulazione sociale — un world model del comportamento umano in senso lato.

6. Scaling e decomposizione della competenza: il debating sulla comprensione

La domanda più profonda che i modelli di linguaggio come world model pongono alla comunità scientifica è quella della comprensione genuina versus la correlazione statistica — riprendendo in chiave linguistica il problema del grounding discusso per i world model visivi. Searle aveva avanzato nel 1980 l'argomento della "stanza cinese" per sostenere che la manipolazione simbolica — per quanto sofisticata — non può dar luogo a comprensione genuina. La risposta dei ricercatori di intelligenza artificiale è cambiata radicalmente nell'era dei grandi modelli di linguaggio.

Un modo di rendere operativa la distinzione tra comprensione e correlazione è attraverso il test di generalizzazione fuori distribuzione. Se un modello comprende il significato di una frase, dovrebbe poter applicare questa comprensione a situazioni nuove che condividono la stessa struttura semantica ma sono espresse in modo superficialmente diverso. Winograd Schemas, HANS, e i dataset di contrast sets testano precisamente questa proprietà — e i modelli di linguaggio mostrano frequenti fallimenti sistematici su questi test, suggerendo una dipendenza da feature superficiali invece della comprensione strutturale.

D'altra parte, la prospettiva funzionalista (Dennett, Hofstadter) sostiene che la comprensione non è una proprietà binaria ma un continuum: un sistema comprende qualcosa nella misura in cui riesce a integrare quella conoscenza coerentemente con il resto della propria base di conoscenza e ad applicarla in modo flessibile a nuovi contesti. In questa prospettiva, GPT-4 mostra una comprensione notevole — sebbene imperfetta — del mondo fisico, sociale e causale, nella misura in cui l'enorme corpus di pre training ha indotto rappresentazioni interne coerenti delle strutture mondane.

Formalmente, il problema può essere posto come la distanza tra il modello implicito del mondo del LLM e il modello del mondo dell'osservatore umano:

$$D(M_{LLM}, M_{human}) = KL[p_{LLM}(W) \| p_{human}(W)]$$

dove $p_{LLM}(W)$ e $p_{human}(W)$ sono le distribuzioni sui possibili stati del mondo indotte rispettivamente dal modello di linguaggio e dall'esperto umano. Questa distanza è minima per le configurazioni mondane ben rappresentate nel corpus (fisica quotidiana, interazioni sociali comuni) e massima per le configurazioni composizionalmente nuove e le situazioni fisiche inusuali. Il programma di ricerca consiste nel ridurre sistematicamente D attraverso pretraining su dati più ricchi, RLHF, retrieval-augmented generation, e — soprattutto — attraverso il grounding nell'esperienza sensorimotoria diretta.

7. Retrieval-Augmented Generation come integrazione di memoria esterna nel world model

I modelli di linguaggio puri soffrono di due limitazioni strutturali rispetto alla funzione di world model: la conoscenza è fissa al momento del pretraining (nessun aggiornamento online), e la capacità della rete di memorizzare fatti specifici è limitata dal numero di parametri. La *Retrieval-Augmented Generation* (RAG, Lewis e collaboratori 2020) supera entrambe le limitazioni dotando il modello di un meccanismo di accesso a una base di conoscenza esterna aggiornabile.

L'architettura RAG si compone di un retriever $p_\eta(z \mid x)$ — tipicamente un modello di recupero denso che codifica la query x in un vettore e recupera i documenti più simili dalla base di conoscenza — e di un generatore $p_\theta(y \mid x, z)$ che condiziona la generazione sulla query x e sui documenti recuperati z. La probabilità della risposta y data la query x è:

$$p_{RAG}(y \mid x) = \Sigma_z \, p_\eta(z \mid x) \, p_\theta(y \mid x, z) \approx \Sigma_{z \in \text{top-k}} \, p_\eta(z \mid x) \, p_\theta(y \mid x, z)$$

dove l'approssimazione somma solo sui top-k documenti recuperati (tipicamente k = 5–10) per rendere il calcolo trattabile. Il training della componente di retrieval può avvenire in modo non supervisionato (dense passage retrieval con negative mining) o in modo end-to-end attraverso REINFORCE o gradient approximations.

Dal punto di vista dei world model, RAG implementa una forma di memoria episodica esterna: la base di conoscenza è un "campione di traiettorie mondane" (documenti, fatti, descrizioni di eventi) che il modello può consultare per aggiornare il proprio prior statistico con informazioni specifiche e recenti. Questa struttura è analoga al replay buffer dei metodi model-based: il world model consulta esperienze passate memorizzate per aggiornare la propria stima dello stato del mondo corrente.

L'aggiornamento della base di conoscenza in RAG corrisponde all'aggiornamento incrementale del modello del mondo in un contesto di apprendimento continuo —

ma senza la catastrofe dell'oblio, poiché i documenti nuovi vengono aggiunti alla base di conoscenza senza sovrascrivere i parametri della rete.

8. Modelli di linguaggio come policy e value function nei sistemi di pianificazione

I modelli di linguaggio non sono solo world model passivi — simulatori del mondo — ma possono essere usati come componenti attivi di sistemi di pianificazione e controllo. In questa funzione, il modello di linguaggio può svolgere tre ruoli distinti che corrispondono alle componenti architetturali dei sistemi di RL model-based.

Nel ruolo di **generatore di piano** (o politica linguistica), il modello di linguaggio riceve la descrizione dello stato corrente e dell'obiettivo e genera un piano in linguaggio naturale — una sequenza di azioni ad alto livello. SayCan (Ahn e collaboratori 2022) implementa questa funzione combinando le probabilità del piano generate dal LLM con una funzione di affordance robotica che valuta la fattibilità di ogni azione nello stato fisico corrente:

a* = argmax_a [log p_LLM(a | goal, history) + log p_afford(a | state)]

Il termine di affordance p_afford(a | state) è appreso da un agente RL e misura la probabilità che l'azione a abbia successo nello stato fisico corrente — ancorando la pianificazione linguistica alla realtà fisica del robot. Senza questo termine, il LLM produrrebbe piani linguisticamente plausibili ma fisicamente impossibili per le capacità specifiche del robot.

Nel ruolo di **funzione di valore linguistica**, il modello di linguaggio valuta l'utilità di stati o di azioni candidate. Nakano e collaboratori (2021) in WebGPT addestrano il modello di linguaggio a scegliere tra diverse azioni di navigazione web (cliccare su un link, formulare una query, citare una fonte) in base al loro valore atteso per la risposta finale — una funzione di valore implicita appresa attraverso feedback umano. Il modello impara a stimare il valore dell'informazione di diverse pagine web per rispondere a una data domanda — una forma di information value nel senso discusso nella pianificazione sotto incertezza.

Nel ruolo di **simulatore del mondo linguistico**, il modello di linguaggio genera scenari ipotetici — "se l'agente eseguisse l'azione a nello stato s, cosa accadrebbe?" — che possono essere usati come rollout impliciti per la pianificazione. La tecnica *Tree of Thoughts* (Yao e collaboratori 2023) implementa esplicitamente questa forma di pianificazione: il modello genera molteplici pensieri (continuazioni parziali) a partire da ogni nodo dell'albero di ricerca, valuta la promettenza di ognuno attraverso un ulteriore prompt di valutazione, e seleziona le continuazioni da esplorare ulteriormente — realizzando in linguaggio naturale una forma di MCTS con il modello stesso come simulatore.

9. RLHF e allineamento dei world model linguistici

Il *Reinforcement Learning from Human Feedback* (RLHF, Christiano e collaboratori 2017; Stiennon e collaboratori 2020; Ouyang e collaboratori 2022) è la procedura di allineamento che ha trasformato i modelli di linguaggio da predittori di token a sistemi in grado di seguire istruzioni, rispettare vincoli etici e produrre output utili e accurati. Dal punto di vista dei world model, RLHF modifica il prior implicito del modello in direzione delle preferenze umane — un aggiornamento bayesiano del prior p_LLM(W) verso distribuzioni più allineate con p_human(W).

La procedura RLHF si articola in tre fasi. La prima è il **supervised fine-tuning** (SFT): il modello base viene fine-tunato su un dataset di dimostrazioni ad alta qualità scritte da esperti umani, producendo il modello SFT M_SFT.

La seconda è l'**addestramento del modello di ricompensa**: per coppie di risposte (y_1, y_2) generate da M_SFT per la stessa query x, un annotatore umano indica quale preferisce. Queste preferenze vengono usate per addestrare una rete di ricompensa $r_\varphi(x, y)$ che modelli la distribuzione delle preferenze umane. Assumendo il modello di preferenza di Bradley-Terry:

$$p(y_1 \succ y_2 \mid x) = \sigma(r_\varphi(x, y_1) - r_\varphi(x, y_2))$$

il training di r_φ minimizza la cross-entropy sulle preferenze annotate:

$$\mathcal{L}_{RM}(\varphi) = -\mathbb{E}_{(x, y_1, y_2, c) \sim \mathcal{D}} [\log p(y_c \succ y_{3-c} \mid x)]$$

dove $c \in \{1, 2\}$ indica la risposta preferita e $\mathcal{D}$ è il dataset di preferenze.

La terza fase è l'**ottimizzazione della politica con RL**: il modello SFT viene ottimizzato per massimizzare il segnale di ricompensa r_φ soggetto al vincolo di non deviare troppo dalla distribuzione di M_SFT — mantenendo la coerenza e la fluidità linguistica mentre si migliora l'allineamento. L'obiettivo è:

$$\max_\pi \mathbb{E}_{x \sim \mathcal{D}, y \sim \pi(\cdot|x)} [r_\varphi(x, y)] - \beta \cdot \mathrm{KL}[\pi(\cdot \mid x) \,\|\, M_{SFT}(\cdot \mid x)]$$

Il termine KL agisce come regolarizzazione: impedisce alla politica ottimizzata π di collassare su un limitato insieme di risposte ad alta ricompensa che possono essere linguisticamente degenerata o fuori distribuzione. Il parametro β controlla il compromesso tra allineamento alle preferenze (minimizzare il primo termine) e fedeltà al modello base (minimizzare la KL).

L'ottimizzazione dell'obiettivo RLHF è tecnicamente delicata perché la KL tra distribuzioni di token su sequenze è non-differenziabile rispetto ai parametri π attraverso il campionamento. Le soluzioni pratiche includono PPO (Proximal Policy Optimization) con clipping del rapporto delle politiche, e la più recente Direct

Preference Optimization (DPO, Rafailov e collaboratori 2023) che sostituisce l'intero loop RL con una loss su coppie di preferenze direttamente sui pesi del modello:

$$\mathcal{L}_DPO(\pi_\theta; M_SFT) = -\mathbb{E}_{(x, y_w, y_l) \sim \mathcal{D}} [\log \sigma(\beta \log(\pi_\theta(y_w|x)/M_SFT(y_w|x)) - \beta \log(\pi_\theta(y_l|x)/M_SFT(y_l|x)))]$$

dove y_w e y_l sono rispettivamente la risposta preferita e quella non preferita. DPO può essere derivato analiticamente come la soluzione in forma chiusa del problema RLHF vincolato alla KL, eliminando la necessità del modello di ricompensa esplicito r_φ e semplificando significativamente il training.

10. Verso modelli linguistici world-model-aware: connessione con le architetture JEPA

La convergenza tra la linea di ricerca sui world model (JEPA, Dreamer, MuZero) e quella sui grandi modelli di linguaggio sta producendo architetture ibride di notevole interesse. Il punto di fusione naturale è la rappresentazione latente condivisa: un modello che rappresenta il mondo sia attraverso token linguistici sia attraverso stati latenti continui potrebbe, in linea di principio, ragionare fluentemente sul mondo fisico usando le stesse rappresentazioni che usa per comprendere e generare linguaggio.

Gato (Reed e collaboratori 2022, Google DeepMind) è il primo tentativo sistematico di unificare testo, immagini, azioni robotiche e osservazioni di giochi in un'unica sequenza di token, processata da un singolo Transformer autoregressivo. Gato codifica azioni continue come token discretizzati, immagini come sequenze di patch tokenizzate attraverso un VQVAE, testo come token BPE — e addestra un singolo Transformer su tutti questi dati eterogenei simultaneamente. Il risultato è un agente generalista che svolge 604 task distinti — chattare, giocare ad Atari, controllare un braccio robotico, descrivere immagini — con una singola rete senza fine-tuning specifico per task.

La limitazione di Gato è che la tokenizzazione discreta di dati continui (azioni, immagini) introduce quantizzazione e perdita di informazione che può essere problematica per task di controllo ad alta precisione. Le architetture più recenti come π_0 (Physical Intelligence 2024) adottano una struttura ibrida: un Transformer di grande dimensione (modello fondazionale linguistico-visivo) fornisce la comprensione semantica e contestuale, mentre una testa di diffusione continua genera le traiettorie di azione senza quantizzazione discreta. La struttura è:

$$(s_context, z_lang) = ViT_LM(osservazione, istruzione) \quad a \sim Diffusion_\pi(a \mid s_context, z_lang)$$

dove ViT_LM è il backbone visio-linguistico fondazionale e $Diffusion_\pi$ è la testa di diffusione che condiziona la generazione dell'azione sullo stato contestuale del backbone. Questo schema combina la ricchezza semantica del pretraining

linguistico-visivo con la precisione e la distribuzione multimodale delle diffusion policy.

La connessione con la JEPA è diretta: il backbone ViT_LM svolge la funzione dell'encoder della JEPA (mappando osservazioni e linguaggio in rappresentazioni latenti predittivamente rilevanti), mentre la diffusion policy svolge la funzione del predittore (generando la sequenza di azioni compatibile con la rappresentazione contestuale). L'integrazione di questi componenti in un unico sistema end-to-end addestrato su dati di interazione robotica reale (dimostrazioni, rollout in simulazione, RLHF) è il fronte di ricerca attivo più promettente per la costruzione di agenti incarnati con autentica comprensione del mondo fisico e sociale.

11. Il problema dell'aggiornamento: conoscenza stagnante e apprendimento continuo dei LLM

Un limite strutturale dei modelli di linguaggio come world model è la staticità della conoscenza: una volta completato il pretraining, la conoscenza del modello rimane congelata al momento del cutoff dei dati. Il mondo fisico e sociale evolve continuamente — nuovi fatti emergono, relazioni cambiano, scoperte scientifiche ridefiniscono la comprensione di fenomeni naturali — e un world model che non si aggiorna è destinato a divergere progressivamente dalla realtà.

Le strategie per affrontare questo problema si articolano su due livelli. Al livello parametrico, il *continual learning* dei LLM mira ad aggiornare i pesi del modello incorporando nuovi dati senza dimenticare la conoscenza precedente. Il problema del catastrophic forgetting — già discusso nel capitolo sull'ottimizzazione — è particolarmente grave per i LLM di grandi dimensioni, poiché i nuovi dati di fine-tuning tendono a sovrascrivere la conoscenza del pretraining in modo sovraproporzionale rispetto alla loro quantità relativa. La regolarizzazione EWC e le sue varianti (FisherMask, WiSE-FT, DARE) proteggono i parametri critici per la conoscenza precedente durante il fine-tuning su nuovi dati.

Al livello non-parametrico, la RAG già discussa permette di aggiornare la conoscenza senza modificare i parametri, aggiungendo documenti alla base di conoscenza recuperabile. Una variante più sofisticata è il *knowledge editing* (Mitchell e collaboratori 2022) — la modifica chirurgica di fatti specifici nei parametri del modello senza riaddestrare l'intera rete. ROME (Rank-One Model Editing) identifica i layer del Transformer che memorizzano un fatto specifico — identificabili attraverso causal tracing che rimuove sistematicamente l'attivazione di ogni layer e misura l'impatto sulla predizione del fatto — e modifica questi layer con un aggiornamento di rango uno:

$W_new = W_old + \Delta$, dove $\Delta = (target - W_old\ k)\ k^T / (k^T k)$

con k la chiave del fatto nel layer identificato e target il nuovo valore del fatto. Questa tecnica permette di modificare fatti specifici (la capitale di un paese, la data di nascita

di una persona) senza impatto misurabile sulla conoscenza circostante — una forma di aggiornamento bayesiano chirurgico del world model parametrico.

L'analogia con i modelli del mondo per il RL è diretta: il knowledge editing corrisponde all'aggiornamento del modello di transizione solo nella regione dello spazio degli stati dove è stata osservata una discrepanza con la realtà — il principio di aggiornamento locale che minimizza il compounding error nell'RSSM e nelle architetture Dreamer.

12. Simulazione sociale computazionale: dalla previsione linguistica alla comprensione istituzionale

La dimensione più ambiziosa dell'uso dei modelli di linguaggio come world model riguarda la simulazione delle strutture sociali, economiche e istituzionali — fenomeni emergenti dal comportamento collettivo di milioni di agenti che interagiscono nel tempo. I modelli di linguaggio addestrati su corpus di testo che includono diari, news, discussioni politiche, testi economici hanno inevitabilmente codificato nelle loro rappresentazioni latenti strutture che riflettono la dinamica sociale.

Argyle e collaboratori (2023) hanno dimostrato che i grandi modelli di linguaggio possono *rispecchiare* le distribuzioni di opinioni politiche di sottogruppi della popolazione americana con notevole accuratezza, se condizionati su descrizioni demografiche appropriate. Il modello non simula l'opinion polling, ma produce output linguistici con distribuzione di preferenze politiche coerente con i dati di sondaggio per il gruppo demografico descritto nel prompt. Questa capacità — denominata *silicon sampling* — apre la strada all'uso dei LLM come simulatori di opinioni per la ricerca in scienze sociali computazionali.

Il framework formale di questi esperimenti è quello della distribuzione condizionale p(opinione | demografica, LLM), che approssima p(opinione | demografica, popolazione_reale) nella misura in cui il corpus di pre training rispecchia le distribuzioni di opinione nella popolazione. Le limitazioni sono evidenti: il corpus non è rappresentativo della popolazione (è biased verso testo prodotto da utenti di internet, di fascia di età e istruzione specifica, in lingue dominanti), e le opinioni estratte dal LLM riflettono le correlazioni statistiche nel corpus piuttosto che la razionalità autonoma degli individui simulati.

Nonostante questi limiti, la simulazione sociale computazionale basata su LLM rappresenta una novità epistemologica: per la prima volta, è possibile simulare il comportamento di grandi popolazioni di agenti con modelli cognitivamente ricchi — invece dei semplici automi con regole fisse della *agent-based modeling* tradizionale — e studiare come le strutture macro-sociali (norme, istituzioni, mercati) emergano dall'interazione di questi agenti. La sfida metodologica consiste nel validare queste simulazioni contro dati empirici reali e nel comprendere in che misura i bias del corpus si propagano nelle strutture emergenti della simulazione — questione che

richiede il quadro epistemologico della causalità strutturale discussa nel capitolo precedente.

XVI. Memoria, architetture episodiche e apprendimento online nei modelli del mondo

1. Il problema della memoria nei sistemi intelligenti: una prospettiva computazionale

Il termine "memoria" designa, tanto nel discorso neuroscientifico sulle strutture cognitive degli organismi biologici quanto in quello ingegneristico sui sistemi artificiali, un insieme di meccanismi funzionalmente distinti che interagiscono in modi non banali. La memoria di lavoro (*working memory*), che mantiene disponibili le rappresentazioni correnti per la manipolazione cognitiva immediata, si distingue dalla memoria episodica (*episodic memory*), che registra eventi specifici con la loro collocazione spaziotemporale, dalla memoria semantica (*semantic memory*), che codifica fatti generali sul mondo, e dalla memoria procedurale (*procedural memory*), che immagazzina le competenze motorie e cognitive acquisite attraverso la pratica. Nei sistemi di apprendimento automatico, queste distinzioni funzionali si traducono in scelte architetturali concrete che hanno implicazioni profonde per le capacità di generalizzazione, adattamento e pianificazione dei modelli del mondo.

La tensione fondamentale che attraversa tutte le architetture di memoria per i sistemi intelligenti è quella tra la *capacità* e la *velocità di accesso*. Le reti neurali profonde, nella loro forma base, codificano la conoscenza nei pesi sinaptici — una forma di memoria distribuita a capacità teoricamente illimitata ma con accesso lento (richiede un nuovo forward pass) e aggiornamento lentissimo (richiede una nuova iterazione di ottimizzazione). Al polo opposto, un registro di accesso rapido con poche posizioni di memoria offre accesso immediato ma capacità limitata. Tra questi estremi si collocano le architetture ibride — reti ricorrenti con stato nascosto, banche di chiavi e valori, grafi di memoria esterna — che cercano di combinare la ricchezza delle rappresentazioni profonde con la flessibilità dell'accesso diretto.

Per i modelli del mondo, la questione della memoria è cruciale per almeno tre ragioni distinte. La prima è la gestione della storia temporale: un agente deve mantenere informazione sugli eventi passati per inferire lo stato del mondo corrente (il POMDP richiede la belief state, che si calcola integrando tutta la storia delle osservazioni). La seconda è la generalizzazione cross-episodica: un agente deve essere in grado di trasferire conoscenza acquisita in un episodio a episodi successivi, senza dover riacquisirla da zero. La terza è la pianificazione a lungo orizzonte: la pianificazione su orizzonti temporali estesi richiede di mantenere traccia degli obiettivi, dei sotto obiettivi e dei vincoli per tempi che eccedono la capacità della finestra di contesto delle architetture standard.

2. Memoria di lavoro nelle reti ricorrenti: LSTM, GRU e i loro limiti

Le reti ricorrenti — LSTM e GRU — implementano una forma di memoria di lavoro attraverso lo stato nascosto $h \in \mathbb{R}^d$, aggiornato a ogni passo temporale da una funzione parametrica degli input correnti e dello stato precedente. Nella LSTM di Hochreiter e Schmidhuber (1997), il meccanismo di gating introduce una separazione tra memoria di breve termine (stato nascosto h) e memoria di lungo termine (cella di memoria c):

$f = \sigma(W_f [h_{-1}; x] + b_f)$ (forget gate) $i = \sigma(W_i [h_{-1}; x] + b_i)$ (input gate) $\tilde{c}_\square = \tanh(W_c [h_{-1}; x] + b_c)$ (candidato cella) $c = f \odot c_{-1} + i \odot \tilde{c}_\square$ (aggiornamento cella) $o = \sigma(W_o [h_{-1}; x] + b_o)$ (output gate) $h = o \odot \tanh(c)$ (stato nascosto)

Il forget gate f controlla quale porzione della cella precedente viene conservata; l'input gate i controlla quale porzione della nuova informazione viene incorporata. Quando $f \approx 1$ e $i \approx 0$, la cella c è quasi invariata — implementando una memoria persistente per quella componente. Quando $f \approx 0$, la cella viene azzerata — dimenticando rapidamente l'informazione precedente.

Il gradiente della loss rispetto alla cella $c_{-\square}$ (n passi nel passato) si propaga attraverso la catena di gate:

$$\partial c / \partial c_{-\square} = \Pi_{=1}^{n} f_{-\square+1}$$

Quando i forget gate sono vicini a 1, questo prodotto rimane O(1) anche per n grande — risolvendo il problema del gradiente che svanisce rispetto alle reti ricorrenti semplici. Tuttavia, quando un forget gate è inferiore a 1 in un certo passo temporale, il contributo all'errore che si propaga oltre quel passo viene attenuato proporzionalmente. Per sequenze molto lunghe (centinaia di migliaia di passi), il segnale di training rimane comunque debole.

Più fondamentalmente, la capacità della memoria di lavoro LSTM è limitata dalla dimensione d dello stato nascosto. La compressione di tutta la storia rilevante in un vettore fisso di dimensione d introduce una perdita di informazione inevitabile quando la struttura della storia è complessa. Per i modelli del mondo che devono tracciare molti oggetti in interazione, molte variabili di stato indipendenti, e strutture relazionali arbitrariamente complesse, questa compressione è un collo di bottiglia fondamentale — non uno che si possa superare aumentando d, poiché l'aumento di d introduce proporzionale aumento del numero di parametri e della complessità computazionale.

3. Memorie esterne differenziabili: Neural Turing Machine e Differentiable Neural Computer

L'architettura Neural Turing Machine (NTM, Graves e collaboratori 2014) ha proposto di superare i limiti della memoria di lavoro ricorrente introducendo una

memoria esterna esplicita e un meccanismo di lettura/scrittura differenziabile. La struttura della NTM è ispirata all'architettura della macchina di Turing: un controller (tipicamente una rete ricorrente) interagisce con una matrice di memoria $M \in \mathbb{R}^{N \times d}$ tramite operazioni di lettura e scrittura parametrizzate da vettori di attenzione.

Il meccanismo di lettura produce un vettore r letto dalla memoria come combinazione convessa delle righe, pesata da un vettore di attenzione $w \in \Delta(N)$:

$r = \Sigma_i \, w_i \, M_i$

dove M_i è la i-esima riga della memoria. Il vettore di attenzione w è calcolato dal controller attraverso un meccanismo che combina indirizzamento per contenuto (content-based addressing) e indirizzamento per posizione (location-based addressing).

L'indirizzamento per contenuto calcola la similarità tra un vettore di ricerca k (prodotto dal controller) e ogni riga della memoria:

w^{content}_i = exp(β C(k, M_i)) / Σ exp(β C(k, M))

dove $C(u, v) = u^T v / (\|u\|_2 \, \|v\|_2)$ è la cosine similarity e β è un parametro di sharpening che controlla la selettività dell'attenzione (β grande → attenzione concentrata sulla riga più simile; β piccolo → attenzione diffusa).

Il meccanismo di scrittura modifica la memoria attraverso un'operazione di erase seguita da un'operazione di add:

$\tilde{M}_i \leftarrow M_i \, [1 - w_i \, e^T]$ (erase, con vettore di cancellazione $e \in [0,1]^d$) $M_i \leftarrow \tilde{M}_i + w_i \, a^T$ (add, con vettore di aggiunta $a \in \mathbb{R}^d$)

Tutte le operazioni sono differenziabili: i vettori di attenzione w, i vettori di ricerca k, i parametri di sharpening β, i vettori e e a sono prodotti dal controller attraverso strati lineari, e i gradienti fluiscono dalla loss attraverso la memoria al controller tramite backpropagation.

Il Differentiable Neural Computer (DNC, Graves e collaboratori 2016) estende la NTM con tre meccanismi aggiuntivi. Il primo è il temporal link matrix $L \in [0,1]^{N \times N}$, che traccia l'ordine temporale in cui le posizioni di memoria sono state scritte, permettendo all'indirizzamento di recuperare sequenze di informazioni nell'ordine in cui sono state memorizzate. Il secondo è il usage vector $u \in [0,1]^N$, che traccia quali posizioni di memoria sono in uso, permettendo al controller di allocare dinamicamente le posizioni libere attraverso l'allocation weighting:

$a_i = (1 - u_i) \, \Pi_{<i} \, u$ (prodotto cumulativo delle usage delle posizioni precedenti)

Il terzo è il meccanismo di read modes che permette al controller di scegliere tra indirizzamento per contenuto, forward traversal (leggere le posizioni nell'ordine temporale in cui sono state scritte) e backward traversal (ordine inverso).

Il DNC ha dimostrato di risolvere task di memorizzazione e recupero che erano inaccessibili alle reti ricorrenti standard: copia di sequenze arbitrarie, ricerca in grafi con struttura di grafo memorizzata nella memoria esterna, completamento di blocchi Sudoku. Per i modelli del mondo, l'architettura DNC è particolarmente rilevante per la memorizzazione esplicita di mappe dell'ambiente — l'equivalente di una "mappa cognitiva" neurale nel senso di Tolman.

4. Memoria episodica per l'apprendimento non-parametrico: nearest-neighbor retrieval

Un'alternativa alla memoria differenziabile è la memoria episodica non-parametrica basata su recupero per similitudine. Invece di aggiornare una matrice di memoria parametrica attraverso backpropagation, questo approccio mantiene un buffer di transizioni passate $\{(s, a, r, s_{+1})\}$ e recupera le più simili alla situazione corrente attraverso k-nearest-neighbor (kNN) search. Il valore della transizione corrente è stimato come media pesata dei valori delle transizioni recuperate:

$\tilde{Q}(s, a) = \Sigma\ w\ Q(s\ , a\ )$

dove s e a sono lo stato e l'azione del k-esimo vicino, e $w \propto 1/d(s, s\ )$ con d una funzione di distanza nello spazio delle rappresentazioni.

Gli *Episodic RL* agents (Blundell e collaboratori 2016, Pritzel e collaboratori 2017) implementano questo schema con un episodic memory module (EMM) che memorizza le esperienze più significative — gli stati ad alta ricompensa o altamente informativi — e le recupera per guidare la pianificazione. Neural Episodic Control (NEC) di Pritzel e collaboratori mantiene una Differentiable Neural Dictionary (DND) per ogni azione, dove le chiavi sono le codifiche degli stati e i valori sono le stime di Q:

$Q_NEC(s, a) = w(s, \mathcal{D}_a)^T v_a$

con $w(s, \mathcal{D}_a) \in \Delta(K)$ il vettore di peso del kNN lookup su $\mathcal{D}_a$ e v_a le stime di valore memorizzate per l'azione a. L'aggiornamento di $\mathcal{D}_a$ avviene scrivendo nuove coppie (h_s, Q_n) dove $h_s = \varphi(s)$ è la codifica dell'encoder e Q_n è il n-step return empirico, e aggiornando le stime già presenti per stati simili.

Per i modelli del mondo, la memoria episodica non-parametrica ha un vantaggio cruciale rispetto alle architetture parametriche: gli aggiornamenti sono immediati (one-shot) e non soggetti a catastrophic forgetting. Quando il modello del mondo visitato uno stato nuovo con dinamica inaspettata, la transizione viene memorizzata nella DND e immediatamente recuperabile alla prossima visita di stati simili — senza la necessità di molte iterazioni di training per incorporare la nuova conoscenza nei pesi della rete.

5. Trasformers con memoria: Transformer-XL, memory-augmented Transformers e Mamba ricorrente

I Transformer standard processano sequenze di lunghezza fissa, determinata dalla finestra di contesto (context window) — tipicamente 512–128.000 token a seconda dell'implementazione. Per i modelli del mondo che devono mantenere coerenza su scale temporali molto superiori a questa finestra — un robot che deve ricordare dove ha posizionato oggetti ore prima, un agente che deve pianificare su scale di giorni — la finestra di contesto finita è un limite fondamentale.

Transformer-XL (Dai e collaboratori 2019) affronta il problema attraverso la *recurrence across segments*: invece di processare ogni segmento di sequenza indipendentemente, le rappresentazioni calcolate per il segmento precedente vengono memorizzate e usate come chiavi e valori aggiuntivi nel meccanismo di attenzione del segmento corrente:

x ^{mem} = StopGradient([h_{τ-1}^{(l)}; h ^{(l-1)}]) q ^{(l)} = h ^{(l-1)} W_Q^{(l)} k ^{(l)} = x ^{mem} W_K^{(l)}, v ^{(l)} = x ^{mem} W_V^{(l)} h ^{(l)} = Attn(q ^{(l)}, k ^{(l)}, v ^{(l)})

dove h_{τ-1}^{(l)} sono le rappresentazioni al layer l del segmento precedente (memorizzate in una cache esterna) e h ^{(l-1)} sono le rappresentazioni del layer precedente del segmento corrente. Lo StopGradient previene la retropropagazione attraverso la cache, rendendola una memoria fissa per il segmento corrente.

Con L layer e memoria di lunghezza M (numero di rappresentazioni memorizzate per segmento), Transformer-XL accede a una finestra di contesto effettiva di L × M token — il campo ricettivo si estende linearmente con la profondità del modello. Per modelli con 24 layer e cache di 512 token per segmento, la dipendenza effettiva si estende a 12.288 token — un ordine di grandezza superiore alla finestra di contesto standard.

Compressive Transformer (Rae e collaboratori 2020) estende Transformer-XL con un secondo livello di memoria: oltre alla cache di memoria recente (le ultime M rappresentazioni), mantiene una memoria compressa (le rappresentazioni più vecchie, compresse attraverso una pooling o convoluzione differenziabile) di dimensione $M_o \geq M$. Quando la memoria recente si riempie, i token più vecchi vengono compressi e spostati nella memoria compressa invece di essere scartati:

M_comp = Compress_φ(M_mem_old)

dove Compress_φ applica una pooling convoluzionale per c token in uno, riducendo la memoria di un fattore c. La memoria compressa viene usata come chiavi e valori aggiuntivi nell'attenzione, con il modello che impara implicitamente a recuperare le informazioni rilevanti dalla memoria compressa quando la query lo richiede.

Per i modelli del mondo video, questa architettura è particolarmente rilevante: i frame video recenti vengono memorizzati ad alta risoluzione nella memoria recente, mentre i frame più lontani vengono compressi (mantenendo solo le caratteristiche semanticamente rilevanti) nella memoria compressa. Il modello può quindi ragionare su dinamiche a lungo termine senza la complessità computazionale dell'attenzione quadratica sull'intera sequenza video.

6. Memoria associativa di Hopfield: la revisione moderna e l'attenzione come recupero

Le reti di Hopfield (Hopfield 1982) sono sistemi di memoria associativa che memorizzano pattern come attrattori di una dinamica di energia. Data una collezione di pattern memorizzati $\{\xi_1, \ldots, \xi \} \subseteq \mathbb{R}^d$, la matrice dei pesi di connessione è:

$$W = (1/d) \Sigma \ \xi \ \xi^T$$

La dinamica di recupero parte da un pattern parziale x_0 (la query "rumorosa") e aggiorna iterativamente:

$$x_{+1} = \text{sign}(W x)$$

convergendo a uno degli attrattori, che corrispondono ai pattern memorizzati (o a loro "stati spurii"). La capacità di memorizzazione delle reti di Hopfield classiche è limitata a circa 0.14 n pattern per n neuroni — un limite di capacità derivabile dall'analisi di campo medio.

Ramsauer e collaboratori (2021) hanno introdotto le *Modern Hopfield Networks*, una generalizzazione che usa un'energia esponenziale invece di quella quadratica classica:

$$E(x) = -\log \Sigma^k \exp(\xi^T x) + (1/2) x^T x + \log n + (d/2) \log(2\pi)$$

La regola di aggiornamento che minimizza questa energia è:

$$x_{+1} = X \ \text{softmax}(\beta X^T x)$$

dove $X = [\xi_1, \ldots, \xi]$ è la matrice dei pattern memorizzati e β è un parametro di temperatura. Per β grande, $\text{softmax}(\beta X^T x)$ è concentrato sul pattern più simile a x , e la regola di aggiornamento recupera esattamente quel pattern — un'operazione di recupero one-shot con capacità esponenziale in d (invece di lineare come nelle reti di Hopfield classiche).

Il collegamento con l'attenzione scalata dei Transformer è immediato e non casuale. Se si identifica $x \equiv Q$ (query), $X \equiv K^T$ (chiavi trasposte), e il mapping finale con $W^T \equiv V$ (valori), la regola di aggiornamento della Modern Hopfield Network diventa:

$$x_{+1} = V \ \text{softmax}(\beta K x) = V \ \text{softmax}(K Q / \beta)$$

che è esattamente l'attenzione scalata con $\beta = 1/\sqrt{d}$. Questo risultato stabilisce che il meccanismo di self-attention dei Transformer è equivalente a un passo di aggiornamento di una Modern Hopfield Network: il Transformer esegue, in ogni layer, un'operazione di recupero associativo dai pattern memorizzati nelle chiavi e valori. La capacità di memorizzazione delle Modern Hopfield Networks è esponenzialmente grande: circa exp(d/2) pattern possono essere memorizzati e recuperati senza interferenza per pattern stocastici gaussiani.

Questa prospettiva unifica l'attenzione e la memoria associativa in un unico formalismo, e suggerisce che i Transformer sono, in senso preciso, memorie associative differenziabili — una computazione di recupero implementata come rete neurale feedforward. La connessione con i modelli del mondo è profonda: ogni strato del Transformer implementa un passo di recupero dalla "base di conoscenza" codificata nei pesi W_K, W_V, recuperando pattern rilevanti per la query corrente.

7. Architetture di memoria per la navigazione: Neural Radiance Fields e mappe metriche implicite

Per i robot che navigano in ambienti fisici, la memoria del mondo assume la forma di una mappa spaziale — una rappresentazione della geometria e semantica dello spazio tridimensionale che il robot ha esplorato. I metodi classici di costruzione della mappa (SLAM tradizionale) usano rappresentazioni esplicite come nuvole di punti, mappe di occupazione o grafi di pose. I metodi neurali recenti costruiscono mappe implicite attraverso funzioni neurali continue.

Le *Neural Radiance Fields* (NeRF, Mildenhall e collaboratori 2020) rappresentano una scena 3D come una funzione continua $F_\theta: \mathbb{R}^3 \times \mathbb{S}^2 \to \mathbb{R}^4$ che mappa ogni punto nello spazio (x, y, z) e ogni direzione di vista (θ, φ) in una coppia (colore RGB, densità volumetrica σ). La rete neurale che implementa F_θ è tipicamente un MLP a 8 layer con skip connections:

$$(\sigma, c) = F_\theta(x, d)$$

dove $x \in \mathbb{R}^3$ è la posizione e $d \in \mathbb{R}^3$ è la direzione di vista normalizzata. La sintesi dell'immagine per una data posa di camera avviene attraverso volume rendering: per ogni raggio $r(t) = o + t\,d$ emanato dalla camera in direzione d, si campionano N punti lungo il raggio e si integra il colore:

$$\hat{C}(r) = \Sigma_{i=1}^{N} T_i \,(1 - \exp(-\sigma_i \,\delta_i))\, c_i, \text{ con } T_i = \exp(-\Sigma <_i \sigma\ \delta\ )$$

dove $\delta_i = t_{i+1} - t_i$ è la distanza tra campioni adiacenti e T_i è la trasmittanza accumulata (probabilità che il raggio non venga assorbito prima del campione i). La loss di training minimizza la differenza tra il colore sintetizzato e quello osservato per ogni raggio dell'insieme di immagini di training:

$$\mathcal{L}_{NeRF} = \Sigma_{r \in R} \|\hat{C}(r) - C(r)\|_2^2$$

Per i modelli del mondo naviganti, la NeRF costituisce una mappa implicita della scena: una volta addestrata, il modello robot può interrogare F_θ per ottenere la geometria in qualsiasi punto dello spazio senza raccogliere nuove osservazioni — una forma di memorizzazione continua della struttura tridimensionale dell'ambiente.

NeRF-SLAM (Rosinol e collaboratori 2023) e GO-SLAM (Zhang e collaboratori 2023) integrano la NeRF con l'ottimizzazione incrementale della posa, permettendo la costruzione online della mappa implicita mentre il robot naviga in un ambiente sconosciuto. Lo stato del sistema è la coppia (parametri della rete F_θ, posa corrente della camera π_t), aggiornata ad ogni frame attraverso:

$$(F_\theta^*, \pi_t^*) = \text{argmin}_{F_\theta, \pi_t} \Sigma_{r \in R_t} \|\hat{C}(r; F_\theta, \pi_t) - C_t(r)\|_2^2$$

L'ottimizzazione è svolta attraverso gradient descent su F_θ e π_t simultaneamente — mappatura e localizzazione ottimizzate congiuntamente attraverso la stessa loss di rendering.

Per i modelli del mondo embodied, la connessione con JEPA è diretta: la NeRF implementa una forma di predizione latente nello spazio 3D — la rappresentazione implicita di densità e colore è la "rappresentazione latente" della scena, e la query attraverso volume rendering è l'equivalente del predittore JEPA che produce stime a partire dalla rappresentazione. La proposta di LeCun di gerarchie JEPA per la navigazione può essere concretizzata attraverso gerarchie di NeRF a diversi livelli di risoluzione: NeRF a bassa risoluzione per la pianificazione del percorso globale, NeRF ad alta risoluzione per la manipolazione locale.

8. Apprendimento online e adattamento continuo: il problema del world model che dimentica

L'apprendimento online — la capacità di aggiornare il modello del mondo in tempo reale man mano che nuove osservazioni arrivano — è una componente essenziale di qualsiasi agente che debba operare in un mondo non stazionario. Gli ambienti fisici cambiano: gli oggetti vengono spostati, nuovi ostacoli compaiono, le proprietà dei materiali variano con le condizioni ambientali (umidità, temperatura). Un modello del mondo che non si aggiorna diventa progressivamente meno accurato, degradando la qualità della pianificazione.

Il problema fondamentale dell'apprendimento online per le reti neurali è il già discusso catastrophic forgetting: aggiornare i pesi della rete sulle osservazioni recenti tende a sovrascrivere la conoscenza acquisita sulle osservazioni passate, producendo un modello accurato per le condizioni correnti ma amnesiaco rispetto alle condizioni precedenti. Per i modelli del mondo, questo si manifesta come una rete che predice correttamente le transizioni nell'ambiente corrente (dopo il cambiamento) ma ha dimenticato come l'ambiente si comportava prima.

Le soluzioni pratiche si articolano su quattro strategie principali.

La **rehearsal** (ripasso) campiona periodicamente dal replay buffer delle osservazioni passate e le include nel batch di aggiornamento insieme alle osservazioni correnti. Sia $\mathcal{B}_old$ il buffer delle osservazioni passate e o l'osservazione corrente; il batch di training è $\mathcal{B}_old \cup \{o\}$, e la loss è:

$$\mathcal{L}_total = \mathcal{L}_world(o) + \lambda_rehearsal \cdot (1/|\mathcal{B}_old|) \Sigma_\{o \in \mathcal{B}_old\} \mathcal{L}_world(o)$$

Il campionamento da $\mathcal{B}_old$ può essere uniforme (Experience Replay) o prioritizzato (Prioritized Experience Replay, dove le osservazioni con errore di predizione più alto vengono campionate con frequenza maggiore).

Il **continual world model learning** (CWML, Kessler e collaboratori 2023) affronta il problema addestrando il modello con un obiettivo che distingue esplicitamente tra componenti della dinamica che cambiano (non-stazionarie) e componenti che rimangono stabili (stazionarie). Per ogni componente latente z dello stato, si stima un indice di non-stazionarietà ϱ che misura quanto la distribuzione di z è cambiata tra episodi recenti e passati:

$$\varrho = KL[p_recent(z) \| p_past(z)]$$

Le componenti ad alto ϱ vengono aggiornate più aggressivamente (learning rate alta) nelle osservazioni recenti, mentre le componenti a basso ϱ vengono aggiornate conservativamente, implementando una forma adattiva di EWC per i componenti del modello del mondo.

Il **meta-continual learning** (He e collaboratori 2019) combina il meta-apprendimento con l'apprendimento continuo: il meta-parametro Θ viene addestrato per essere un buon punto di partenza non solo per l'adattamento rapido a nuovi task, ma anche per la resistenza al forgetting durante l'apprendimento sequenziale. L'outer loop del meta-training include task simulati di learning sequenziale, e il meta-learner viene valutato sulla sua capacità di mantenere le prestazioni sui task precedenti mentre apprende il task corrente.

La **online distillation** (Rebuffi e collaboratori 2017 per iCaRL, Li e Hoiem 2016 per LwF) usa una copia frozen del modello precedente come "insegnante" durante il training sul dato corrente:

$$\mathcal{L}_distillation = \Sigma_k (f_prev(x) - f_curr(x))^2 + \mathcal{L}_new_task(x)$$

dove f_prev è la copia frozen del modello precedente e f_curr è il modello corrente. Il termine di distillazione forza il modello corrente a produrre output simili a quelli del modello precedente per i dati del replay buffer, preservando la conoscenza precedente mentre si apprende la nuova.

9. Memoria gerarchica e pianificazione a lungo orizzonte: il problema della composizione temporale

La pianificazione a lungo orizzonte — su scale temporali di ore, giorni o settimane — richiede di mantenere traccia di obiettivi, sotto obiettivi, vincoli e risorse a diverse granularità temporali, coordinando livelli di astrazione diversi in una struttura gerarchica coerente. La *Hierarchical Reinforcement Learning* (HRL) formalizza questo requisito attraverso architetture a più livelli in cui politiche di alto livello selezionano sotto obiettivi, e politiche di basso livello implementano le azioni primitive necessarie per raggiungere ogni sotto obiettivo.

Il framework *Options* di Sutton, Precup e Singh (1999) formalizza le azioni temporalmente estese attraverso la nozione di *opzione*: una tripla $(\mathcal{I}, \pi, \beta)$ dove $\mathcal{I} \subseteq \mathcal{S}$ è l'insieme di stati in cui l'opzione può essere iniziata, π è la politica interna che governa il comportamento durante l'esecuzione dell'opzione, e $\beta: \mathcal{S} \to [0,1]$ è la funzione di terminazione che specifica la probabilità di terminare l'opzione in ogni stato. La politica di alto livello seleziona tra opzioni invece che tra azioni primitive, e viene aggiornata attraverso l'SMDP (Semi-Markov Decision Process) che l'esecuzione delle opzioni induce.

La funzione di valore dell'opzione o iniziata nello stato s è:

$$V_\pi^o(s) = \mathbb{E}_\pi [\Sigma_{t=0}^{T-1} \gamma^t r_{+1} \mid s_0 = s, \text{option } o]$$

dove T è il tempo di terminazione dell'opzione (stocastico se β non è una funzione indicatrice). Le equazioni di Bellman per la SMDP estendono quelle del MDP includendo il valore del sotto-policy durante l'esecuzione dell'opzione:

$$Q^{\Omega}(s, o) = \Sigma_{s'} \Sigma_{t=1}^{\infty} P(s = s', \tau = t \mid s_0 = s, o) [r + \gamma^t V^{\Omega}(s')]$$

dove Ω è l'insieme delle opzioni disponibili e $V^\Omega(s) = \max_{o \in \Omega, o \text{ iniziabile in } s} Q^\Omega(s, o)$.

Per i modelli del mondo gerarchici, l'architettura di memoria deve supportare stati goal a diversi livelli temporali. In HIRO (Hierarchical RL with Off-policy Correction, Nachum e collaboratori 2018), la politica di alto livello produce sotto obiettivi nello spazio delle osservazioni $g \in \mathcal{S}$, e la politica di basso livello massimizza la reward intrinseca di avvicinarsi al sotto obiettivo:

$$r_i(s, g, s_{+1}) = -\| \varphi(s) + g - \varphi(s_{+1}) \|_2$$

dove φ è una funzione di codifica delle osservazioni. Il modello del mondo di basso livello predice s_{+1} dato s e l'azione primitiva a; il modello del mondo di alto livello predice il prossimo sotto obiettivo g_{t+c} dopo c passi primitivi, condizionato su s e sulla scelta del sotto-obiettivo corrente g. La memoria del sistema mantiene sia il

sotto obiettivo corrente g (che rimane fisso per c passi primitivi) sia la storia degli stati visitati (per valutare il progresso verso g).

Per orizzonti estremamente lunghi — del tipo necessario per la pianificazione di task domestici complessi come "organizzare una festa" o "preparare un viaggio" — la struttura gerarchica delle opzioni non è sufficiente. Servono architetture di memoria che possano mantenere rappresentazioni di obiettivi ad alta astrazione per periodi arbitrariamente lunghi, aggiornandoli selettivamente quando le condizioni del mondo cambiano in modo rilevante per l'obiettivo. I *Large Language Model agents* con *scratchpad* e *memory streams* (Park e collaboratori 2023) implementano questa funzione attraverso la memoria esplicita in linguaggio naturale: i ricordi passati vengono periodicamente riassunti (compressi semanticamente, analogamente al Compressive Transformer) e recuperati attraverso ricerca per rilevanza semantica quando serve pianificare.

10. Memoria di lavoro e bias cognitivi: connessioni con la neuroscienza computazionale

La neuroscienza computazionale offre modelli quantitativi delle architetture di memoria biologiche che forniscono vincoli e ispirazioni per i sistemi artificiali. La corteccia prefrontale (PFC) è stata identificata come il substrato neurale principale della memoria di lavoro negli esseri umani e nei primati: essa mantiene rappresentazioni attive delle informazioni contestuali rilevanti per il task corrente attraverso il rinforzo persistente delle attività neurali (attività sostenuta, *persistent activity*).

Il modello a variabile di stato (*state variable model*, Miller, Lundqvist e Bhatt 2018) formalizza la PFC come un sistema dinamico con due regimi: un regime di codifica rapida (*coding*), in cui le rappresentazioni vengono scritte nella PFC attraverso input afferenti, e un regime di mantenimento (*maintenance*), in cui le rappresentazioni vengono sostenute attraverso attività ricorrente interna. Il passaggio tra i due regimi è controllato da modulazione dopaminergica, che funge da "gate" biologico — un meccanismo che risuona con il forget gate e l'input gate delle LSTM.

Il modello di Baddeley e Hitch della memoria di lavoro distingue l'*esecutivo centrale* (che coordina le risorse cognitive), il *loop fonologico* (che mantiene informazioni verbali nell'immediato), il *taccuino visuo-spaziale* (che mantiene immagini e layout spaziali), e il *buffer episodico* (che integra informazioni da diverse fonti in una rappresentazione coerente). Questa architettura modulare è sorprendentemente congruente con le architetture Transformer multi-head: diverse teste di attenzione possono specializzarsi nel mantenimento di diversi tipi di informazione contestuale, con la proiezione finale che integra i contributi di tutte le teste in una rappresentazione coerente — esattamente la funzione del buffer episodico di Baddeley.

La memoria a lungo termine nell'ippocampo, e in particolare il meccanismo di consolidazione della memoria (*memory consolidation*) che avviene durante il sonno, ha ispirato l'architettura di *Experience Replay* nei sistemi di RL. L'ippocampo registra

rapidamente ogni episodio in una forma episodica specifica (come la memoria episodica non-parametrica discussa in precedenza); durante il sonno, le esperienze vengono "replicate" (*replay*) alla corteccia, permettendo la graduale incorporazione nei pesi corticali (che corrispondono alla memoria semantica a lungo termine). La separazione temporale tra acquisizione rapida (ippocampo) e consolidazione lenta (corteccia) previene l'interferenza tra i due sistemi — esattamente il problema che il catastrophic forgetting nella rete neurale unica non riesce a evitare.

Il *complementary learning systems* (CLS, McClelland e collaboratori 1995) formalizza questa struttura come due sistemi di apprendimento complementari: uno rapido e specifico (ippocampo → replay buffer, kNN memory), uno lento e generale (corteccia → parametri della rete neurale). L'interazione tra i due sistemi — il sistema rapido invia segnali di replay al sistema lento, che generalizza attraverso l'integrazione su molti episodi — è il principio fondante del Dyna di Sutton e della prioritized experience replay moderna. Le architetture che implementano esplicitamente questa separazione — come le Episodic RL con DND per l'acquisizione rapida e rete Q per la generalizzazione — mostrano empiricamente una migliore efficienza del campionamento e una minore fragilità rispetto alle reti singole.

11. Sparse autoencoders e rappresentazioni interpretabili per i world model

Un filone di ricerca recente e promettente per l'analisi e il miglioramento dei modelli del mondo riguarda l'uso di *sparse autoencoders* (SAE) per decomporre le rappresentazioni intermedie dei modelli in componenti interpretabili. L'ipotesi di linearità delle rappresentazioni (*linear representation hypothesis*) sostiene che i concetti semanticamente distinti sono codificati come direzioni lineari nello spazio delle attivazioni — una singola dimensione, o una combinazione lineare di poche dimensioni, corrisponde a un concetto o proprietà specifica.

Elhage e collaboratori (2022) formalizzano questa ipotesi come *superposition hypothesis*: i modelli neurali codificano più feature di quante siano le dimensioni dell'attivazione, attraverso la sovrapposizione di vettori quasi-ortogonali in spazi di alta dimensione. Se d è la dimensione dello spazio di attivazione e n >> d è il numero di feature rilevanti per il task, il modello rappresenta ogni feature f come un vettore $u_f \in \mathbb{R}^d$ e l'attivazione $a \in \mathbb{R}^d$ per un input x è approssimata come:

$$a(x) \approx \Sigma_f \alpha_f(x)\, u_f$$

dove $\alpha_f(x) \in \mathbb{R}$ è il "grado di attivazione" della feature f sull'input x. I vettori $\{u_f\}$ non sono ortogonali (non ci sarebbe spazio per tutti), ma sono quasi-ortogonali: $u_f \cdot u_g \approx 0$ per $f \neq g$. Questo schema è possibile in alta dimensione per il Johnson-Lindenstrauss lemma: n vettori quasi-ortogonali possono essere embedded in $\mathbb{R}^d$ con $|u_f \cdot u_g| \leq \varepsilon$ per $n \leq e^{O(\varepsilon^2 d)}$.

Un *sparse autoencoder* è una rete con un bottleneck ampiamente sovradimensionato:

$z(a) = \mathrm{ReLU}(W_{enc}\, a + b_{enc})$ (codifica sparsa, $z \in \mathbb{R}^M$ con $M >> d$) $\hat{a}(a) = W_{dec}\, z(a) + b_{dec}$ (ricostruzione) $\mathcal{L}_{SAE} = \|a - \hat{a}\|_2^2 + \lambda \|z\|_1$ (ricostruzione + sparsità L1)

La perdita L_1 sulla codifica z incoraggia la sparsità: la maggior parte delle componenti di z è zero per ogni input, e solo poche (le feature attivate) sono non-nulle. Le colonne di W_{dec} corrispondono ai vettori di feature $\{u_f\}$: la ricostruzione $\hat{a} \approx \Sigma_f\, z_f\, W_{dec,f}$ decompone l'attivazione nella somma delle feature attivate.

Applicando i SAE alle attivazioni intermedie di modelli di linguaggio e world model, Cunningham e collaboratori (2023) e Bricken e collaboratori (2023) hanno identificato migliaia di feature monosemantiche — feature che si attivano per concetti specifici e interpretabili: il token "Paris" in contesti geografici, il concetto di "velocità" in descrizioni fisiche, il "gioco degli scacchi" in contesti di gioco. Per i modelli del mondo, questa tecnica di decomposizione potrebbe permettere di identificare le feature che codificano le proprietà fisicamente rilevanti dello stato del mondo — posizione degli oggetti, velocità, tipo di materiale — separandole dalle feature che codificano proprietà stilistiche irrilevanti per la pianificazione.

La connessione con la JEPA è diretta: il predittore della JEPA, per essere efficace, deve apprendere a operare sulle feature semanticamente rilevanti (quelle che sono predittive del target) ignorando le feature irrilevanti. Un SAE addestrato sulle attivazioni dell'encoder della JEPA potrebbe rendere esplicite queste feature, fornendo interpretabilità alla struttura del world model latente e permettendo di verificare empiricamente se le rappresentazioni apprese corrispondono a concetti fisicamente significativi.

12. Verso una teoria unificata della memoria per i modelli del mondo

Riassumendo il percorso di questo capitolo, si può identificare una struttura concettuale coerente che attraversa le diverse architetture di memoria discusse. Tutte le soluzioni al problema della memoria nei world model condividono una struttura tripartita: un meccanismo di *scrittura* (come le nuove esperienze vengono incorporate nella struttura di memoria), un meccanismo di *lettura* (come le esperienze passate rilevanti vengono recuperate quando serve), e un meccanismo di *dimenticanza selettiva* (come le informazioni obsolete o irrilevanti vengono progressivamente deprioritizzate).

Il meccanismo di scrittura nelle reti ricorrenti è la regola di aggiornamento della cella di memoria, governata dai gate di input e forget; nelle reti con memoria esterna è l'operazione di write della NTM/DNC o l'inserimento nella DND del kNN; nei Transformer con cache è l'aggiornamento della cache dei layer rappresentativi; nelle NeRF è l'ottimizzazione dei pesi della rete sull'insieme di osservazioni correnti.

Il meccanismo di lettura è l'applicazione dello stato nascosto come vettore di query al meccanismo di attenzione nello stato corrente; il kNN lookup nella DND; l'attenzione

al contenuto della cache nel Transformer-XL; il volume rendering che interroga la NeRF attraverso ray marching.

Il meccanismo di dimenticanza selettiva è il forget gate nell'LSTM (dimenticanza rapida, adattiva al contenuto); il decay del vettore di usage nel DNC (le posizioni non lette vengono gradualmente rilasciate); la LRU eviction nella cache del Transformer; la ri-ottimizzazione della NeRF su nuovi frame (le informazioni implicite nei pesi vengono aggiornate verso le nuove osservazioni).

La teoria unificante che connette tutte queste architetture è quella dell'inferenza variazionale sequenziale: ogni meccanismo di memoria implementa, in un'approssimazione più o meno accurata, l'aggiornamento bayesiano della distribuzione di credenza sullo stato del mondo $p(s \mid o_1, \ldots, o)$. Le differenze tra le architetture riflettono diverse scelte sul tipo di prior adottato (gaussiano, categoriale, non-parametrico), sulla famiglia di distribuzioni approssimanti (gaussiana variazionale, particelle, kNN), e sulla struttura della verosimiglianza (ricorrente, attenzionale, radiante). Questa prospettiva unificante è già presente in forma implicita nell'RSSM di Dreamer — che distingue esplicitamente lo stato deterministico (la "cache" della storia passata) dallo stato stocastico (la distribuzione di credenza sull'incertezza irriducibile) — e il suo sviluppo sistematico verso architetture più ricche e scalabili è il programma di ricerca più ambizioso nel campo dei modelli del mondo per l'intelligenza artificiale autonoma.

XVI. Modelli generativi avanzati come modelli del mondo: flussi normalizzanti, mappe di trasporto e modelli di score

1. Il problema della densità esplicita e le sue implicazioni per la transizione di stato

Nei capitoli precedenti si è visto come i modelli a variabile latente di tipo VAE-RSSM affrontino il problema della modellazione delle distribuzioni di transizione attraverso un'approssimazione variazionale che introduce per costruzione un gap tra la vera posteriore e la sua approssimazione gaussiana. Questo gap non è una debolezza accidentale ma una conseguenza strutturale della scelta di famiglie variazionali che bilancino la trattabilità computazionale con la flessibilità rappresentativa. Esiste tuttavia un'intera famiglia di architetture generative — i *normalizing flows*, i modelli di score, e le mappe di trasporto ottimale — che permettono di definire distribuzioni di probabilità con densità esplicita e arbitrariamente complessa su spazi continui, senza ricorrere all'approssimazione variazionale. La loro integrazione nei modelli del mondo apre prospettive teoriche e pratiche di grande interesse, e merita una trattazione formale autonoma.

Il punto di partenza è la domanda fondamentale: cosa si intende per modello di transizione con densità esplicita? Un modello di transizione p_θ(s_{t+1} | s_t, a_t) possiede una densità esplicita se, per ogni coppia (s_{t+1}, s_t, a_t), è possibile calcolare p_θ(s_{t+1} | s_t, a_t) esattamente in tempo polinomiale nel numero di parametri. Questa proprietà è banale per modelli gaussiani (la densità gaussiana ha forma chiusa) ma non triviale per distribuzioni multimodali, a code pesanti, o con supporto su varietà di dimensione inferiore allo spazio ambiente. Un modello di transizione con densità esplicita permette tre operazioni che sono precluse o approssimate nei modelli VAE: la stima della log-verosimiglianza esatta di una traiettoria osservata, il calcolo dei gradienti della log-verosimiglianza per massima verosimiglianza diretta, e la valutazione della probabilità di stati specifici per la pianificazione ottimale.

2. Flussi normalizzanti: costruzione e formalismo del cambio di variabile

I *normalizing flows* (NF) costruiscono distribuzioni complesse come trasformazioni biettive di distribuzioni semplici. Sia $z \in \mathbb{R}^d$ una variabile con distribuzione base $p_z(z)$ — tipicamente gaussiana isotropa $\mathcal{N}(0, I)$ — e sia $f_\theta: \mathbb{R}^d \to \mathbb{R}^d$ una biiezione differenziabile con inversa differenziabile (diffeomorfismo). La variabile trasformata $x = f_\theta(z)$ ha densità:

$$p_x(x) = p_z(f_\theta^{-1}(x)) \cdot \left|\det J_{f_\theta^{-1}}(x)\right|$$

dove $J_{f_\theta^{-1}}(x) = \partial f_\theta^{-1}/\partial x \in \mathbb{R}^{d \times d}$ è la matrice jacobiana dell'inversa calcolata in x. Equivalentemente, usando il diffeomorfismo inverso $z = f_\theta^{-1}(x)$:

$$\log p_x(x) = \log p_z(z) - \log \left|\det J_{f_\theta}(z)\right|$$

dove $J_{f_\theta}(z) = \partial f_\theta / \partial z$ è la jacobiana della mappa diretta. La log-densità di x è quindi la log-densità base valutata nel punto di provenienza, corretta dal log-determinante jacobiano — il termine che quantifica la dilatazione o contrazione di volume indotta dalla mappa f_θ.

La composizione di K trasformazioni $f^1_\theta \circ f^2_\theta \circ \cdots \circ f^K_\theta$ produce un flusso con densità:

$$\log p_x(x) = \log p_z(z_0) - \sum_{k=1}^{K} \log \left|\det J_{f^k_\theta}(z_{k-1})\right|$$

dove $z_0 = (f^K_\theta)^{-1} \circ \cdots \circ (f^1_\theta)^{-1}(x)$ e $z_k = f^k_\theta(z_{k-1})$. Con trasformazioni sufficientemente espressive e catene sufficientemente lunghe, i flussi normalizzanti possono approssimare arbitrariamente bene qualsiasi distribuzione assolutamente continua — un risultato analogo al teorema di approssimazione universale per le funzioni, derivabile dalla continuità della topologia della convergenza debole sulle distribuzioni di probabilità.

Il requisito di calcolabilità del log-determinante jacobiano è il vincolo architetturale principale che distingue le diverse famiglie di flussi. Per una matrice densa d×d generica, il calcolo del determinante ha complessità $O(d^3)$ — inaccettabile per spazi ad alta dimensione come le immagini ($d \sim 10^4 - 10^6$). Le soluzioni proposte nella letteratura si articolano in tre strategie principali.

La prima è la struttura triangolare del jacobiano. Se f_θ è una trasformazione affine con jacobiana triangolare, il determinante è il prodotto degli elementi diagonali — complessità O(d). Le *coupling layers* (Dinh e collaboratori, 2014, 2017 con NICE e RealNVP) costruiscono trasformazioni con questa struttura partizionando le variabili in due gruppi (x_1, x_2) e definendo:

$$x_2' = x_2 \odot \exp(s_\theta(x_1)) + t_\theta(x_1) \; x_1' = x_1$$

dove s_θ e t_θ sono reti neurali arbitrarie che agiscono solo sul primo gruppo. La jacobiana di questa trasformazione è triangolare a blocchi con blocchi diagonali $\{1, \exp(s_\theta(x_1))\}$, e il suo determinante è $\Sigma_i \, s_\theta(x_1)_i$ — una somma scalare indipendente dalla dimensione di x_2. L'inversa è esplicitamente calcolabile: $x_2 = (x_2' - t_\theta(x_1')) \odot \exp(-s_\theta(x_1'))$. Alternando il ruolo di x_1 e x_2 tra gli strati del flusso, si ottiene una trasformazione globale che modifica tutte le componenti.

La seconda strategia è la struttura fattorizzata. Il Glow (Kingma e Dhariwal, 2018) combina coupling layers con *1×1 convoluzioni invertibili* — le cui jacobiane sono matrici quadrate di dimensione pari al numero di canali (tipicamente 3-512) con determinante calcolabile tramite decomposizione LU — e *operazioni di actnorm* (normalizzazione dell'attivazione per canale con parametri di scala e traslazione appresi). Questa composizione produce un flusso con densità esplicita su immagini ad alta risoluzione, capace di generare campioni di qualità fotografica per sintesi condizionata.

La terza strategia, più recente, è la struttura continua. I *Continuous Normalizing Flows* (CNF, Chen e collaboratori 2018) abbandonano la struttura discreta a strati e definiscono il flusso come la soluzione di un'ODE nel tempo fittizio $t \in [0, 1]$:

$$dz/dt = f_\theta(z(t), t), \; z(0) = z_0$$

La log-densità evolve lungo il flusso secondo la *formula di Liouville* (o dell'istanza della traccia):

$$d \log p(z(t))/dt = -\mathrm{tr}(\partial f_\theta/\partial z(t))$$

dove $\mathrm{tr}(\partial f_\theta/\partial z)$ è la traccia della jacobiana — calcolabile in O(d) invece di $O(d^3)$ usando il trucco dello Hutchinson estimator: $\mathrm{tr}(A) = E_{\varepsilon \sim N(0,I)}[\varepsilon^T A \varepsilon]$, stimato con un campione singolo con varianza finita. La log-densità dell'output z(1) è quindi:

$$\log p(z(1)) = \log p(z(0)) - \int_0^1 \mathrm{tr}(\partial f_\theta/\partial z(t)) \, dt$$

questa formula è la realizzazione continua del cambio di variabile e può essere integrata numericamente insieme alla dinamica z(t) con un singolo solver ODE,

sfruttando il metodo aggiunto per il gradiente (come discusso nel capitolo sui Neural ODE).

3. Flussi normalizzanti come modelli di transizione: architetture e vincoli

L'integrazione dei flussi normalizzanti nei modelli del mondo richiede di costruire un flusso condizionale: invece di modellare p(s_{t+1}), si modella p(s_{t+1} | s_t, a_t) come un flusso la cui distribuzione base e/o le cui trasformazioni dipendono dal contesto (s_t, a_t). L'architettura più naturale è quella del *conditioned normalizing flow*:

$z \sim N(0, I)$ $s_{t+1} = f_\theta(z; s_t, a_t)$

dove $f_\theta(\cdot; c)$ è un diffeomorfismo condizionato sul contesto $c = (s_t, a_t)$. La densità della transizione è quindi:

$$\log p_\theta(s_{t+1} \mid s_t, a_t) = \log N(z; 0, I) - \log \left|\det J_{f_\theta}(z; s_t, a_t)\right|$$

con $z = f_\theta^{-1}(s_{t+1}; s_t, a_t)$. Il condizionamento può essere implementato in modi diversi: i parametri di ogni strato del flusso possono essere generati da una rete di *hypernetwork* che prende in input il contesto; il contesto può essere concatenato all'input di ogni coupling layer; o può essere usato come variabile condizionante nelle trasformazioni affini (come in RealNVP condizionato).

Un vantaggio cruciale dei flussi condizionali come modelli di transizione è la *esatta valutazione della verosimiglianza*. Dato un dataset di traiettorie reali $\{(s_t, a_t, s_{t+1})\}$, il modello può essere addestrato per massima verosimiglianza diretta:

$$\mathcal{L}_{MLE} = -E_{(s_t, a_t, s_{t+1}) \sim D} [\log p_\theta(s_{t+1} \mid s_t, a_t)]$$

senza introdurre il gap dell'approssimazione variazionale. Nella pratica dell'RSSM, questo gap può essere significativo — la divergenza KL tra prior e posterior nella loss dell'ELBO agisce come un compromesso tra la precisione del posterior (che usa l'osservazione reale) e la qualità del prior (che deve generare traiettorie coerenti senza l'osservazione). I flussi eliminano questo compromesso: il modello di transizione ha densità esplicita e può essere ottimizzato direttamente sulla log-verosimiglianza delle transizioni osservate.

Il rovescio della medaglia è il costo computazionale. Il calcolo del log-determinante jacobiano nelle coupling layers richiede due forward pass (uno per la trasformazione, uno per l'inversa) invece di uno, e il CNF continuo richiede l'integrazione dell'ODE — un overhead di 2-10× rispetto a un forward pass di rete gaussiana. Per gli RSSM su task di controllo con spazi di stato di dimensione moderata ($d \sim 32$–256), questo overhead è accettabile; per modelli su immagini ad alta risoluzione ($d \sim 4096+$), può diventare proibitivo.

4. Flow Matching: la frontiera della semplicità computazionale

Il *Flow Matching* (FM, Lipman e collaboratori 2022; Albergo e Vanden-Eijnden 2022; Liu, Gong e Liu 2022) ha rivoluzionato l'addestramento dei flussi normalizzanti continui attraverso un obiettivo di training radicalmente più semplice e computazionalmente efficiente rispetto alla massimizzazione della log-verosimiglianza del CNF.

L'intuizione di base è la seguente: invece di imparare un diffeomorfismo che trasformi la distribuzione base nella distribuzione target, si apprende direttamente il *campo vettoriale* v_θ(x, t) tale che il flusso indotto da dz/dt = v_θ(z(t), t) trasporti la distribuzione base p_0 = N(0, I) nella distribuzione target p_1 = p_data a t = 1. Il campo vettoriale ottimale che compie questo trasporto, ammesso che le traiettorie corrispondano alle *interpolazioni lineari* tra campioni della base e campioni del target, è:

v^star(x, t) = E[x_1 - x_0 | x_t = x]

dove x_t = (1-t) x_0 + t x_1 è l'interpolazione lineare, x_0 ~ p_0, x_1 ~ p_1. Questo risultato, denominato *Conditional Flow Matching* (CFM), suggerisce l'obiettivo di training:

$\mathcal{L}$_CFM(θ) = E_{t ~ U[0,1], x_0 ~ p_0, x_1 ~ p_1} [‖v_θ(x_t, t) - (x_1 - x_0) ‖ $_2^2$]

dove il campo vettoriale θ viene addestrato a predire la direzione di trasporto della singola traiettoria lineare (x_0, x_1), che è nota analiticamente come (x_1 - x_0). La semplicità di questo obiettivo è sorprendente: non richiede la traccia jacobiana del CNF, non richiede la retropropagazione attraverso il solver ODE, ed è bilanciato rispetto al tempo t senza bisogno di schedule di rumore. Il gradiente della loss è calcolabile con un singolo forward pass su v_θ e la retropropagazione standard — una complessità analoga all'addestramento di qualsiasi rete neurale.

La connessione con i modelli di diffusione è immediata: i DDPM (Denoising Diffusion Probabilistic Models, Ho e collaboratori 2020) con la loro formulazione in termini di score matching possono essere visti come un caso speciale di Flow Matching con traiettorie non lineari (che seguono lo schedule di rumore gaussiano). Il Flow Matching generalizza questo schema a traiettorie arbitrarie e mostra che le traiettorie lineari (Optimal Transport conditioned FM) producono traiettorie più dirette, richiedendo meno step di integrazione per raggiungere campioni di qualità simile.

Per i modelli del mondo, il Flow Matching si presta a costruire un modello di transizione che apprende v_θ(x, t; s_t, a_t) — il campo vettoriale condizionato sul contesto — e genera s_{t+1} integrando numericamente da x_0 ~ N(0, I) a x_1 = s_{t+1} lungo la traiettoria di trasporto. La qualità delle previsioni è comparabile ai modelli di diffusione, il costo di campionamento è ridotto di 2-5× grazie alle traiettorie più rettilinee, e il training è più stabile per la semplicità dell'obiettivo.

5. Modelli di score e diffusione come world model: DDPM, DDIM e score-based SDE

I modelli di diffusione (Ho e collaboratori 2020; Song e collaboratori 2020, 2021) hanno prodotto le immagini sintetiche di qualità più alta mai generate da sistemi neurali, e la loro applicazione ai modelli del mondo — come simulatori di transizioni di stato in spazi di osservazione ad alta dimensione — ha prodotto architetture come UniSim e SORA, già discusse nel capitolo sulle leggi di scala. Qui si affrontano le fondamenta matematiche in dettaglio.

Il processo di diffusione *forward* aggiunge progressivamente rumore gaussiano a un campione di dati $x_0 \sim p_{data}(x)$ attraverso una catena di Markov:

$$q(x_t \mid x_{t-1}) = \mathcal{N}(x_t; \sqrt{1-\beta_t}\, x_{t-1}, \beta_t I)$$

con uno schedule di rumore $0 < \beta_1 < \beta_2 < \cdots < \beta_T < 1$. Definendo $\bar{\alpha}_t = \prod_{i=1}^{t} (1-\beta_i)$, il marginale ha forma chiusa:

$$q(x_t \mid x_0) = \mathcal{N}(x_t; \sqrt{\bar{\alpha}_t}\, x_0, (1-\bar{\alpha}_t) I)$$

Per T sufficientemente grande (tipicamente $T = 1000$) e schedule appropriato, x_T è approssimativamente gaussiano isotropo. Il processo *inverso* è la denoising chain:

$$p_\theta(x_{t-1} \mid x_t) = \mathcal{N}(x_{t-1}; \mu_\theta(x_t, t), \Sigma_\theta(x_t, t))$$

Usando il lemma di Bayes e la proprietà del processo forward, la media ottimale del reverse step è:

$$\tilde{\mu}_t(x_t, x_0) = \frac{\sqrt{\bar{\alpha}_{t-1}}\, \beta_t}{1-\bar{\alpha}_t} x_0 + \frac{\sqrt{1-\beta_t}(1-\bar{\alpha}_{t-1})}{1-\bar{\alpha}_t} x_t$$

Ho e collaboratori parametrizzano $x_0 = \frac{x_t - \sqrt{1-\bar{\alpha}_t}\, \varepsilon_\theta(x_t, t)}{\sqrt{\bar{\alpha}_t}}$, dove ε_θ è una rete che predice il rumore aggiunto. Sostituendo:

$$\mu_\theta(x_t, t) = \frac{1}{\sqrt{1-\beta_t}} \left(x_t - \frac{\beta_t}{\sqrt{1-\bar{\alpha}_t}}\, \varepsilon_\theta(x_t, t)\right)$$

La loss di training DDPM minimizza l'errore di predizione del rumore:

$$\mathcal{L}_{DDPM} = \mathbb{E}_{t, x_0, \varepsilon} \left[\| \varepsilon - \varepsilon_\theta(\sqrt{\bar{\alpha}_t}\, x_0 + \sqrt{1-\bar{\alpha}_t}\, \varepsilon, t) \|_2^2\right]$$

La connessione con la teoria dello score matching (Song e collaboratori 2020) rivela la struttura teorica sottostante. Lo *score* di una distribuzione $p_t(x)$ è il gradiente della sua log-densità: $s(x, t) = \nabla_x \log p_t(x)$. *Si dimostra che il modello addestrato con la loss DDPM soddisfa* $\varepsilon_\theta(x_t, t) \approx -\sqrt{1-\bar{\alpha}_t}\, \nabla_{x_t} \log p_t(x_t)$, ossia predice (a meno di un fattore di scala) lo score della distribuzione al passo t. Song e collaboratori generalizzano questo risultato al dominio continuo attraverso la *stochastic differential equation* (SDE) di diffusione:

$$dx = f(x, t)\, dt + g(t)\, dW_t \quad \text{(forward SDE)}$$

La distribuzione p_t(x) al tempo t evolve secondo l'equazione di Fokker-Planck. La SDE inversa (che scorre indietro nel tempo per generare campioni) è:

$$dx = [f(x, t) - g(t)^2 \nabla_x \log p_t(x)] dt + g(t) dW_t$$

dove dW_t è un moto browniano retrograde. Sostituendo il modello di score approssimato $s_\theta(x, t) \approx \nabla_x \log p_t(x)$, si ottiene la SDE generativa approssimata — che può essere integrata numericamente con solver SDE efficienti (Euler-Maruyama, Heun, DPM-Solver) per campionare dalla distribuzione target.

Il *DDIM* (Denoising Diffusion Implicit Models, Song, Meng e Ermon 2021) reinterpreta la catena di denoising come un campionamento deterministico da un'ODE corrispondente — la *probability flow ODE*:

$$dx = [f(x, t) - (g(t)^2/2) \nabla_x \log p_t(x)] dt$$

che ha le stesse marginali $p_t(x)$ della SDE ma non ha componente stocastica. La campionamento attraverso la probability flow ODE è deterministica: dato $x_T \sim N(0, I)$, l'integrazione dell'ODE fino a $t=0$ produce un campione $x_0 \sim p_{data}(x)$. DDIM permette di ridurre il numero di step di integrazione da $T=1000$ a 20-50 senza degradazione significativa della qualità — un fattore di accelerazione di 20-50× rispetto ai DDPM.

6. Modelli di diffusione come modelli del mondo: architettura, training e inferenza

L'impiego di modelli di diffusione come modelli di transizione nei world model introduce sfide specifiche che vanno oltre il semplice condizionamento del denoiser su (s_t, a_t). La sfida principale è quella dell'inferenza rapida: un pianificatore che deve valutare migliaia di traiettorie candidate (come in CEM o MPPI) necessita di campionare da $p(s_{t+1} \mid s_t, a_t)$ molte volte in rapida successione. I 20-50 step necessari per DDIM, moltiplicati per le migliaia di rollout e l'orizzonte di pianificazione, producono un budget computazionale proibitivo.

Le soluzioni proposte nella letteratura si articolano in tre direzioni. La prima è la *distillazione del modello di diffusione* in un modello a un passo (consistency models, Song e collaboratori 2023). Un *consistency model* $f_\theta(x_t, t)$ apprende a mappare direttamente qualsiasi punto della traiettoria di diffusione x_t nella sua sorgente pulita x_0:

$$\mathcal{L}_{consistency} = E [d(f_\theta(x_{t_n}, t_n), f_{\theta^-}(x_{t_{n-1}}, t_{n-1}))]$$

dove f_{θ^-} è una copia EMA del modello e d è una misura di distanza. L'addestramento forza la consistenza del modello lungo la traiettoria: ogni punto proiettato deve dare lo stesso output, indipendentemente dal punto di partenza. Un consistency model addestrato campiona con un singolo forward pass — eliminando completamente il costo dell'integrazione ODE per scenari dove la qualità del singolo passo è sufficiente.

La seconda direzione è la *latent diffusion* (Rombach e collaboratori 2022). Invece di eseguire la diffusione nello spazio dei pixel (alta dimensione), un autoencoder VQ-VAE o KL-VAE comprime l'osservazione in uno spazio latente di dimensione molto inferiore (tipicamente d_latent ~ d_pixel / 16 – 64), e il modello di diffusione opera in questo spazio latente. Il costo computazionale scala con d_latent invece che con d_pixel, permettendo di mantenere qualità di generazione elevata a costo ridotto. Per i world model, la latent diffusion fornisce un modello di transizione nello stesso spazio latente usato dall'encoder dell'RSSM o della JEPA — integrando naturalmente con le architetture esistenti.

La terza direzione è il *conditioned deterministic approximation*: una rete deterministica apprende la media della distribuzione di transizione E[s_{t+1} | s_t, a_t], usata per la pianificazione, mentre un modello di diffusione residuale modella la distribuzione dell'errore s_{t+1} - E[s_{t+1} | s_t, a_t] — che ha scala molto più piccola e richiede meno step di denoising per essere campionata accuratamente. Questo schema divide la modellazione in due componenti con requisiti computazionali diversi: la componente deterministica (economica) gestisce la maggior parte della varianza delle transizioni, mentre la componente diffusiva (costosa) modella l'incertezza residua, campionata solo quando necessario.

7. Trasporto ottimale come prior per la transizione di stato

Il trasporto ottimale, la cui geometria è stata introdotta nel capitolo sulla geometria riemanniana, offre un'ulteriore prospettiva sul problema della modellazione della transizione. La mappa di trasporto ottimale $T^: \mathbb{R}^d \to \mathbb{R}^d$ *che trasporta la distribuzione degli stati al tempo t, p(s_t), nella distribuzione degli stati al tempo t+1, p(s_{t+1}), con costo minimo di trasporto quadratico, è la mappa di Brenier:* $T^ = \nabla \varphi$, dove φ è una funzione convessa (il potenziale di Kantorovitch). Questa mappa ha proprietà geometriche uniche: è la trasformazione tra distribuzioni che minimizza il displacement medio $E[\|s_{t+1} - s_t\|^2]$, ossia il trasporto più diretto tra le due distribuzioni.

L'Entropic Optimal Transport (EOT) regolarizza il piano di trasporto con un termine di entropia:

$$W_{\varepsilon}(p, q) = \min_{\gamma \in \Gamma(p,q)} \iint \|x - y\|^2 \, d\gamma(x, y) + \varepsilon \, KL[\gamma \,\|\, p \otimes q]$$

dove la regolarizzazione entropia $\varepsilon > 0$ rende il problema strettamente convesso e ammette una soluzione unica con la struttura di Schrödinger bridge: il piano ottimale γ^* ha la forma:

$$d\gamma^*(x, y) = \exp((f(x) + g(y) - \|x-y\|^2/\varepsilon) / 1) \, dp(x) \, dq(y)$$

dove f e g sono le potenziali duali di Sinkhorn, calcolabili iterativamente attraverso il noto algoritmo di Sinkhorn-Knopp. Per $\varepsilon \to 0$, il piano di Sinkhorn converge al piano di trasporto ottimale; per $\varepsilon \to \infty$, converge al prodotto $p \otimes q$ (trasporto indipendente, senza preferenza per la corrispondenza).

Lo *Schrödinger Bridge* (SB) generalizza il trasporto ottimale al caso delle traiettorie stocastiche: invece di trovare la mappa di trasporto ottimale, si cerca il processo stocastico più vicino (in KL divergenza) al moto browniano standard che marginalizza a p_0 per t=0 e a p_1 per t=1. Il risultato è una SDE:

$$dx = [u^*(x, t) + \sigma^2/2 \nabla_x \log \psi(x, t)] dt + \sigma dW_t$$

dove u^* è il campo vettoriale di trasporto e ψ è la soluzione di un problema di Schrödinger — una coppia di equazioni alle derivate parziali paraboliche accoppiate. Vargas e collaboratori (2021) e De Bortoli e collaboratori (2021) mostrano come stimare la soluzione dello SB attraverso campionamento iterativo, ottenendo un modello di transizione con traiettorie ottimali nel senso del trasporto entropico.

Per i modelli del mondo, il trasporto ottimale offre un prior naturale sulla struttura delle transizioni: le traiettorie reali del mondo fisico sono — per il principio di minima azione della meccanica classica — percorsi che minimizzano un funzionale d'azione. Il trasporto ottimale, minimizzando il costo di spostamento quadratico, realizza esattamente questo principio per dinamiche con energia cinetica proporzionale a $\|v\|^2$. L'integrazione tra Schrödinger Bridge e architetture RSSM, dove il prior sullo stato latente sostituisce il prior gaussiano isotropo standard con la distribuzione di SB, è una direzione di ricerca attiva che promette modelli di transizione con migliore generalizzazione fisica.

8. Modelli autoregressivi per spazi di stato discreti: VQVAE e trasformatori generativi

Nei capitoli precedenti si è visto che IRIS e Genie discretizzano lo spazio degli stati attraverso VQVAE (Vector Quantized VAE) e modellano la transizione nello spazio dei token discreti attraverso un Transformer autoregressivo. Questa architettura costituisce in sé un *modello generativo* con densità esplicita sugli spazi discreti — la probabilità di una sequenza di token $s_{1:T}$ è la produzione delle probabilità condizionali:

$$\log p_\theta(s_{1:T}) = \Sigma_{t=1}^{T} \log p_\theta(s_t \mid s_{<t})$$

dove $p_\theta(s_t \mid s_{<t})$ è la distribuzione categoriale prodotta dall'attenzione causale del Transformer sul contesto passato. La massima verosimiglianza su questo obiettivo è equivalente alla minimizzazione della cross-entropy, che è l'obiettivo standard per i modelli linguistici autoregressivi.

La quantizzazione vettoriale nel VQVAE introduce una discontinuità nella mappa codifica → token che rende il gradiente nullo quasi ovunque. Il *straight-through estimator* di Bengio e collaboratori (2013) gestisce questa discontinuità propagando il gradiente del decoder attraverso la quantizzazione come se fosse l'identità:

$$\nabla_{z_e} \mathcal{L} \approx \nabla_{z_q} \mathcal{L} \text{ (straight-through)}$$

dove z_e è la codifica continua pre-quantizzazione e z_q è il codebook vector selezionato. Questo estimatore è distorto ma ha varianza bassa in pratica, e produce codici discreti di qualità superiore agli estimatori di Gumbel-Softmax (che hanno varianza alta per temperature basse). Il VQVAE addestrato su immagini o frame video produce tipicamente K×K token per frame (ad esempio 16×16 = 256 token per frame 256×256 con patch-size 16), ciascuno estratto da un codebook di N vettori (tipicamente N = 512–8192).

L'equivarianza del Transformer autoregressivo allo spazio discreto dei token VQVAE produce un modello di transizione con proprietà teoriche interessanti. La densità del modello è definita sulla distribuzione congiunta sullo spazio dei token, che è un simplesso di dimensione N^K - 1 per K token per frame — uno spazio di dimensione molto più alta del corrispondente spazio degli stati continuo. Questa scelta aumenta la flessibilità ma richiede un numero di parametri molto maggiore per coprire adeguatamente la distribuzione. Il vantaggio principale è la compatibilità con i meccanismi di attenzione e con il pre training su grandi corpora testuali — permettendo il trasferimento di conoscenza dal dominio del linguaggio al dominio visivo attraverso sequenze di token misti.

9. Valutazione dei modelli generativi come world model: metriche e benchmark

La valutazione dei modelli generativi impiegati come world model richiede metriche che vadano oltre la qualità percettiva delle osservazioni sintetizzate e catturino le proprietà fisicamente rilevanti delle previsioni. Le metriche standard per la valutazione dei modelli generativi — FID (Fréchet Inception Distance), IS (Inception Score), KID (Kernel Inception Distance) — misurano la qualità distribuzionale delle immagini generate rispetto a immagini reali, ma non misurano la correttezza fisica delle transizioni simulate.

La FID è definita come la distanza di Fréchet tra due distribuzioni gaussiane fitted sulle feature dell'Inception-v3 dell'insieme reale e generato:

$$FID(p, q) = \|\mu_p - \mu_q\|_2^2 + tr(\Sigma_p + \Sigma_q - 2(\Sigma_p \Sigma_q)^{1/2})$$

dove μ_p, Σ_p e μ_q, Σ_q sono le medie e covarianze delle distribuzioni di feature sull'insieme reale e generato rispettivamente. La formula si riduce alla distanza W_2^2 tra le due gaussiane fitted — una connessione con la geometria del trasporto ottimale discussa in precedenza. Una FID bassa indica che la distribuzione delle immagini generate è simile a quella delle immagini reali nello spazio delle feature semantiche di Inception, ma non implica correttezza causale o fisica delle singole transizioni.

Per valutare la correttezza fisica, le metriche proposte nella letteratura dei world model includono: il *prediction error* sul test set — $\|\hat{s}_{t+1} - s_{t+1}\|_2^2$ mediato su traiettorie reali — che misura la precisione della previsione media ma non cattura la qualità della distribuzione; il *rollout return* — il ritorno cumulativo ottenuto da una politica

addestrata nel modello simulato e valutata nell'ambiente reale — che misura la qualità del world model per la pianificazione ma richiede una politica addestrata; la *physical consistency score* (proposta da PhyGenBench, Wu e collaboratori 2024) — la frequenza con cui le transizioni simulate rispettano le leggi fisiche di conservazione (energia, momento, simmetrie del corpo rigido) — che misura la plausibilità fisica ma richiede un simulatore fisico come oracolo.

Una metrica teoricamente fondata per la qualità di un modello di transizione come strumento di pianificazione è il *simulation lemma* di Kakade e Langford (2002), che stabilisce che se il modello di transizione ha errore di distribuzione ε_model, la politica ottimale nel modello ha una perdita di valore rispetto all'ottimo reale di al più:

$$|V^{*}(s) - V^{\hat{}\pi}(s)| \leq 2\,\gamma\,\varepsilon_\text{model}\,R_\text{max} / (1-\gamma)^2$$

dove R_max è il reward massimo e γ il fattore di sconto. Questa disuguaglianza formalizza il sim-to-real gap per i modelli di transizione e suggerisce che la metrica rilevante non è la FID o il PSNR, ma la divergenza distribuzionale ε_model tra la distribuzione delle transizioni simulate e quelle reali — misurata idealmente in divergenza di total variation o in distanza di Wasserstein per le proprietà di continuità rispetto al reward.

10. Composizione di modelli generativi: mixture of experts e prodotto di esperti

Un'ulteriore direzione di ricerca nei world model generativi riguarda la composizione di molteplici modelli parziali, ciascuno specializzato su una regione o una modalità dello spazio degli stati, in un modello globale coerente. Le due architetture principali di composizione sono la *Mixture of Experts* (MoE) e il *Product of Experts* (PoE).

Nella MoE, la distribuzione di transizione è una mistura pesata di K modelli esperti, ognuno responsabile di una regione dello spazio degli stati:

$$p_\theta(s_\{t+1\} \mid s_t, a_t) = \Sigma_\{k=1\}^\{K\}\ \pi_k(s_t, a_t)\ p_k(s_\{t+1\} \mid s_t, a_t)$$

dove $\pi_k(s_t, a_t) \in \Delta(K)$ sono i pesi del gating network e p_k sono le distribuzioni esperto. Ogni esperto può essere un flusso normalizzante, un modello gaussiano, o un modello di diffusione, e il gating network decide quale esperto è più competente per lo stato e l'azione correnti. La MoE è particolarmente adatta per ambienti con transizioni qualitativamente diverse — ad esempio, un robot in ambienti con superfici diverse (liscia, ruvida, deformabile) che hanno dinamiche di contatto radicalmente diverse.

Il PoE definisce la distribuzione congiunta come prodotto normalizzato di K distribuzioni parziali:

$$p_\theta(s_\{t+1\} \mid s_t, a_t) \propto \Pi_\{k=1\}^\{K\}\ p_k(s_\{t+1\} \mid c_k(s_t, a_t))$$

dove c_k è il k-esimo contesto (una proiezione o trasformazione dell'input). Il PoE ha la proprietà che ogni esperto agisce come vincolo sulla distribuzione: la distribuzione finale è concentrata nelle regioni dove tutti gli esperti assegnano alta probabilità — una logica di AND logistico invece dell'OR della MoE. Hinton (2002) ha proposto il PoE come modello per le distribuzioni con struttura combinatoria — dove la distribuzione globale è vincolata da molti fattori indipendenti che devono essere soddisfatti simultaneamente. Per i world model, il PoE è naturale per modellare vincoli fisici multipli che devono essere rispettati contemporaneamente: conservazione dell'energia, non-interpenetrazione degli oggetti, rispetto dei limiti articolari del robot.

La normalizzazione del PoE richiede il calcolo della funzione di partizione $Z = \int \Pi_k p_k(x \mid c_k)\, dx$, che è in generale intrattabile. L'*Energy-Based Model* (EBM) discusso nel capitolo sulla JEPA è formalmente equivalente a un PoE con energia $E(x) = -\Sigma_k \log p_k(x \mid c_k)$ — e le stesse tecniche di addestramento (contrastive divergence, score matching, noise contrastive estimation) si applicano. Questo rivela una connessione profonda tra la JEPA come EBM e i world model generativi basati su PoE: l'energia della JEPA $F_{JEPA}(x, y, z) = D(h_\varphi(f_\theta(x), z), g_\psi(y))$ può essere interpretata come l'energia negativa di un PoE dove ogni termine della distanza D è un fattore del prodotto.

11. Inferenza ammortizzata rapida: encoder di flusso e codificatori normalizzanti

Un'ulteriore sinergia tra flussi normalizzanti e world model riguarda l'inferenza rapida dello stato latente da nuove osservazioni — il problema dell'*encoding*. Nei VAE, l'encoder è una rete che produce i parametri (μ, σ) della gaussiana approssimante — un'approssimazione che introduce sistematicamente un gap rispetto alla vera posteriore. I *normalizing flow encoders* sostituiscono la gaussiana approssimante con un flusso condizionale:

$$q_\varphi(z \mid x) = N(\tilde{z}; 0, I)\ |\det J_{f^{-1}_\varphi}(z)|, \text{ con } z = f_\varphi(\tilde{z}; x)$$

dove $f_\varphi(\cdot; x)$ è un diffeomorfismo parametrizzato dalla rete encoder che dipende dall'input x. Questo schema permette all'encoder di produrre distribuzioni posteriori arbitrariamente complesse — multimodali, a code pesanti, con dipendenze tra coordinate — preservando la trattabilità della log-verosimiglianza. L'ELBO con encoder a flusso è:

$$\mathcal{L}_{flow_VAE} = E_{q_\varphi(z|x)}[\log p_\theta(x \mid z)] - KL[q_\varphi(z \mid x) \,\|\, p(z)] = E_{\tilde{z} \sim N(0,I)}[\log p_\theta(x \mid f_\varphi(\tilde{z}; x))] - E_{\tilde{z} \sim N(0,I)}[\log |\det J_{f_\varphi}(\tilde{z}; x)| + \log N(\tilde{z}; 0, I) - \log p(f_\varphi(\tilde{z}; x))]$$

dove il termine KL si calcola esattamente usando il cambio di variabile del flusso. Per i world model, questo schema permette di approssimare la vera posteriore $q(z_t \mid h_t, o_t)$ dell'RSSM con un flusso condizionale — riducendo il gap variazionale e

migliorando la qualità della stima dello stato latente, specialmente in regime di osservazioni ambigue o parzialmente occlude.

12. Connessione tra modelli generativi avanzati e la JEPA: previsione vs. generazione

La tensione fondamentale tra l'approccio JEPA (predizione nello spazio latente) e l'approccio generativo (modellazione della distribuzione sullo spazio degli input) non è una dicotomia assoluta ma un continuum di scelte architetturali con diversi trade-off. In un estremo del continuum vi è la ricostruzione pixel-perfetta del MAE, che massimizza la fedeltà della previsione nello spazio degli input al costo di allocare capacità a dettagli irrilevanti. All'altro estremo vi è la JEPA pura, che predice nel solo spazio latente e rinuncia completamente alla ricostruibilità delle osservazioni. I modelli generativi avanzati — flussi, score models, SB — occupano posizioni intermedie.

Il *conditional generation* di uno stato futuro condizionato su un encoding latente del contesto è formalmente equivalente a una JEPA con decoder probabilistico: il predittore della JEPA produce uno stato latente $\hat{s}_y \approx s_y$, e il decoder (un flusso o un modello di diffusione) campiona osservazioni plausibili dato $\hat{s}_y$. Questa architettura combina i vantaggi dell'approccio JEPA (predizione compatta e semanticamente ricca nello spazio latente) con quelli dell'approccio generativo (capacità di simulare osservazioni realistiche per la pianificazione visiva o per la valutazione umana delle previsioni). Il costo aggiuntivo è il decoder generativo, che deve essere sufficientemente espressivo da recuperare la variabilità non catturata nel latente.

Il *latent JEPA* con denoising decoder (proposto in DINO v2 e esplorato in I-JEPA con decoder allegato) implementa esattamente questa struttura: l'encoder e il predittore operano nella JEPA, mentre un decoder di diffusione leggermente supervisionato traduce le previsioni latenti in previsioni visive plausibili. La loss totale combina la loss JEPA (nel latente) con la loss di diffusione (nei pixel):

$$\mathcal{L}_{total} = \lambda_{JEPA}\, \mathcal{L}_{JEPA}(\hat{s}_y, s_y) + \lambda_{diff}\, \mathcal{L}_{diffusion}(o, Dec(\hat{s}_y))$$

Il bilanciamento tra i due termini controlla quanto la rete alloca capacità alla semantica (termine JEPA) rispetto ai dettagli percettivi (termine di diffusione). Per i world model orientati alla pianificazione, $\lambda_{JEPA} >> \lambda_{diff}$; per i world model orientati alla simulazione visiva realistica, il rapporto si inverte.

La convergenza tra questi due paradigmi — predizione latente e generazione probabilistica degli input — sembra destinata a produrre architetture ibride che sfruttino i vantaggi di entrambi: la ricchezza semantica delle rappresentazioni latenti della JEPA per la pianificazione efficiente, e la plausibilità visiva dei modelli di diffusione per la valutazione umana e per il grounding percettivo del sistema. Questa integrazione è oggi il fronte di ricerca più attivo nel campo dei world model visivi per la robotica incarnata.

XVI. Grounding multimodale e JEPA visio-linguistica: verso la comprensione integrata del mondo

1. La crisi del grounding e la multimodalità come soluzione strutturale

Il problema del grounding simbolico — la connessione tra le rappresentazioni interne di un sistema artificiale e le entità del mondo fisico che esse denotano — è stato posto con la massima acutezza da Harnad (1990) come estensione critica dell'argomento della stanza cinese di Searle. Harnad osservava che un sistema che manipola simboli riferendosi esclusivamente ad altri simboli non può mai uscire da questo circolo autoreferenziale: il significato dei simboli deve in qualche misura ancorarsi all'esperienza sensoriale diretta. La formulazione originale di Harnad faceva riferimento alla modalità verbale e percettiva come unica via d'uscita dal regresso simbolico; la ricerca contemporanea ha generalizzato questa intuizione in modo radicale, mostrando che la connessione tra linguaggio e percezione visiva — il grounding multimodale — è non solo sufficiente a produrre rappresentazioni semanticamente ricche, ma costituisce probabilmente il meccanismo principale attraverso cui i grandi modelli fondazionali acquisiscono la loro apparente comprensione del mondo.

La prospettiva JEPA, come delineata nel capitolo dedicato, affronta il problema del grounding da una direzione inversa rispetto ai modelli linguistici: invece di ancorare il linguaggio alla percezione, àncora la percezione alla struttura predittiva latente. Un encoder visivo che apprende a predire le rappresentazioni di regioni oscurate della stessa immagine sviluppa rappresentazioni che catturano la struttura semantica della scena — perché la struttura semantica è l'unica informazione predittivamente rilevante per la previsione latente. Tuttavia, questa struttura è puramente visiva e priva di connessione con il linguaggio che gli esseri umani usano per ragionare sulla scena, comunicare i propri piani, e formulare obiettivi complessi. La *JEPA visio-linguistica* è l'architettura che cerca di unificare i due livelli di rappresentazione — visivo e linguistico — in un unico framework di predizione latente congiunto.

La necessità di questa unificazione non è puramente teorica. I robot che operano in ambienti umani devono ricevere istruzioni in linguaggio naturale, interpretarle in relazione al contesto visivo percepito, pianificare sequenze di azioni fisiche per eseguire le istruzioni, e comunicare lo stato del piano agli esseri umani in linguaggio naturale. Nessuno di questi compiti è accessibile a un sistema con rappresentazioni puramente visive o puramente linguistiche; tutti richiedono la capacità di mappare coerentemente tra i due spazi di rappresentazione, in modo che le entità denotate dal linguaggio corrispondano alle regioni o agli oggetti identificati nella scena visiva, e viceversa.

2. CLIP: allineamento contrastivo di immagini e testo

Il *Contrastive Language-Image Pre-Training* (CLIP, Radford e collaboratori 2021, OpenAI) ha stabilito il paradigma dominante per l'apprendimento di rappresentazioni visio-linguistiche congiunte attraverso l'allineamento contrastivo su coppie (immagine, testo). Il dataset di pretraining di CLIP consiste di 400 milioni di coppie (immagine, didascalia) raccolte da internet — un corpus che cattura la distribuzione naturale delle associazioni linguistico-visive prodotte dagli esseri umani in contesti comunicativi reali.

L'architettura CLIP comprende due encoder indipendenti: un encoder visivo f_v (originalmente un ResNet o un ViT) che mappa immagini in un vettore $z_v \in \mathbb{R}^d$, e un encoder testuale f_t (un Transformer causale) che mappa sequenze di testo in un vettore $z_t \in \mathbb{R}^d$. Entrambi i vettori sono proiettati in uno spazio comune di dimensione d attraverso proiezioni lineari e normalizzati alla sfera unitaria. La loss di addestramento è la loss InfoNCE bidimensionale (simmetrica tra modalità) su batch di N coppie:

$$\mathcal{L}_CLIP = -(1/2N)\ \Sigma_i\ [\log \exp(z_{vi} \cdot z_i / \tau) / \Sigma\ \exp(z_{vi} \cdot z\ / \tau) + \log \exp(z_{vi} \cdot z_i / \tau) / \Sigma\ \exp(z_{v\square} \cdot z_i / \tau)]$$

dove τ è un parametro di temperatura appreso (inizialmente $\tau = 0.07$) e le due somme sono rispettivamente il termine image-to-text (la i-esima immagine deve avere alta similarità con la sua didascalia e bassa con le altre N-1 didascalie) e il termine text-to-image (simmetrico). Il fattore $1/\tau$ è equivalente al parametro β di sharpening della softmax: temperature basse producono distribuzione di attenzione più concentrata, incoraggiando l'allineamento preciso tra coppie positive e la separazione dalle negative.

La connessione con la teoria dell'informazione è la stessa dell'InfoNCE nel capitolo sull'SSL: la loss CLIP massimizza un lower bound all'informazione mutua I(Z_v; Z_t) tra le rappresentazioni visiva e testuale delle stesse coppie:

$$I(Z_v; Z_t) \geq \log N - \mathcal{L}_CLIP$$

L'informazione mutua I(Z_v; Z_t) misura la quantità di informazione condivisa tra rappresentazione visiva e testuale — che corrisponde alla struttura semantica della coppia (immagine, testo) che è rilevante per entrambe le modalità. Massimizzarla costringe gli encoder a estrarre, dalle rispettive modalità, le caratteristiche che sono co-occorrenti e mutuamente predittive — precisamente la struttura semantica che rende la coppia semanticamente coerente.

L'aspetto più sorprendente di CLIP è l'emergenza di capacità di zero-shot classification: dato un set di classi $\{c_1, ..., cK\}$, si costruiscono descrizioni testuali del tipo "a photo of a {c }", si codificano con f_t in vettori {z }, e si classifica una nuova immagine x come quella classe con il più alto coseno di similarità tra f_v(x) e {z }. Su ImageNet, il CLIP ViT-L/14 ottiene il 75.5% di accuratezza top-1 in zero-shot —

senza mai aver visto un esempio di ImageNet durante il training — avvicinandosi alla performance di un ResNet-101 supervisionato su tutto il dataset. Questo risultato evidenzia come la co-occorrenza visivo-linguistica a scala internet sia sufficiente a costruire rappresentazioni con struttura semantica profondamente strutturata.

3. La geometria dello spazio visio-linguistico: isomorfismo e gap modale

Nonostante il successo empirico di CLIP, la struttura geometrica dello spazio congiunto visio-linguistico presenta anomalie teoricamente rilevanti. L'analisi della geometria delle rappresentazioni di CLIP rivela che le modalità visiva e testuale occupano regioni distinte dello spazio comune: le rappresentazioni visive tendono a formare cluster densi nello spazio di embedding, mentre le rappresentazioni testuali sono più diffuse e più vicine all'origine. Questo fenomeno, denominato *modality gap* (Liang e collaboratori 2022), implica che la distanza coseno tra rappresentazioni della stessa modalità è in media inferiore alla distanza tra rappresentazioni di modalità diverse, anche per coppie semanticamente allineate.

Il modality gap può essere formalizzato attraverso il concetto di *cono modale*: le rappresentazioni di una stessa modalità tendono a concentrarsi in un cono dallo spazio di embedding, con la norma delle proiezioni ortogonali sul cono comune piccola rispetto alla norma totale. Formalmente, sia $V \subseteq \mathbb{R}^d$ il sottospazio modale visivo (il sottospazio approssimato della distribuzione di rappresentazioni visive) e $T \subseteq \mathbb{R}^d$ quello testuale. Il gap modale è misurato dalla distanza tra i centroidi dei coni:

$$\text{gap_modal} = \| E[z_v] / \|E[z_v]\| - E[z_t] / \|E[z_t]\| \|_2$$

Liang e collaboratori mostrano che il gap modale non è un artefatto del training ma emerge necessariamente dalla geometria della loss InfoNCE con negative sampling uniforme: i campioni negativi sono sistematicamente all'interno della stessa modalità, spingendo le rappresentazioni dello stesso tipo a separarsi da quelle dell'altra modalità. La soluzione proposta — campionamento di negativi inter-modali — riduce il gap ma introduce sfide computazionali nella costruzione dei batch di training.

Il gap modale ha implicazioni pratiche per l'uso di CLIP come backbone nei world model: la distanza tra le rappresentazioni visiva e testuale di una coppia semanticamente allineata non è zero ma un vettore costante (approssimativamente il centro del gap). Le applicazioni di generazione guidata da testo (come Stable Diffusion che usa le rappresentazioni testuali di CLIP come condizionamento) compensano implicitamente questo gap attraverso la cross-attention — che non richiede che le rappresentazioni siano nello stesso punto dello spazio, ma solo che siano in corrispondenza via prodotto scalare. Per le architetture che usano le rappresentazioni CLIP come stati del world model su cui pianificare, il gap può introdurre errori sistematici nella misurazione della distanza semantica tra stati visivi e descrizioni linguistiche degli obiettivi.

4. ALIGN, Florence e SigLIP: varianti e miglioramenti dell'allineamento contrastivo

La famiglia di modelli che estende il paradigma CLIP comprende diverse varianti con contributi tecnici distinti. ALIGN (Jia e collaboratori 2021, Google) scala il corpus a 1.8 miliardi di coppie noisy image-text e dimostra che la scala compensa parzialmente la qualità del dato — con un corpus abbastanza grande di coppie raccolte senza filtraggio, i modelli raggiungono qualità comparabile a CLIP addestrato su dati più curati ma meno numerosi. Questa osservazione ha implicazioni per i world model che potrebbero sfruttare stream continui di dati video non annotati — la varianza del singolo campione è alta, ma il segnale cumulativo su miliardi di campioni è sufficiente per l'apprendimento.

Florence (Yuan e collaboratori 2021, Microsoft) estende l'allineamento contrastivo a un framework gerarchico che include non solo coppie globali (immagine intera, didascalia globale) ma anche allineamenti locali (regione dell'immagine, frase nominale nella didascalia che denota quella regione). L'allineamento locale è realizzato attraverso una loss di grounding contrastiva:

$\mathcal{L}_grounding = -\Sigma_\{(r, p) \in$ coppie positive$\}$ log exp(f_v(r) · f_t(p) / τ) / $\Sigma_\{p' \in$ batch$\}$ exp(f_v(r) · f_t(p') / τ)

dove r è la rappresentazione di una regione dell'immagine (prodotta da una ROI pooling o da una slot attention) e p è la rappresentazione della frase nominale corrispondente. Questo allineamento locale costruisce un mapping più fine-grained tra entità visive e referenti linguistici — avvicinandosi al vero grounding simbolico nel senso di Harnad.

SigLIP (Zhai e collaboratori 2023, Google) sostituisce la loss InfoNCE di CLIP con una loss di sigmoid binary cross-entropy applicata indipendentemente a ogni coppia:

$\mathcal{L}_SigLIP = -\Sigma_i$ [y_i log σ(z_{vi} · z / τ + b) + (1-y_i) log(1 - σ(z_{vi} · z / τ + b))]

dove y_i = 1 se (i, j) è una coppia positiva e 0 altrimenti, σ è la sigmoide, e b è un bias di inizializzazione negativo per compensare lo sbilanciamento tra positive e negative. La loss di sigmoid non richiede la normalizzazione softmax sull'intero batch — eliminando la dipendenza dalla dimensione del batch e permettendo un training efficiente anche con batch piccoli. Questa proprietà è cruciale per i world model dove le traiettorie di training hanno lunghezze variabili e l'allocazione di memoria GPU è una risorsa scarsa.

5. La JEPA visio-linguistica: previsione congiunta nello spazio cross-modale

Il framework JEPA esteso alla multimodalità visio-linguistica costruisce un sistema di previsione che opera non solo all'interno di una modalità (predire patch visive da

patch visive, come in I-JEPA) ma anche tra modalità (predire rappresentazioni testuali da input visivi e viceversa). L'architettura risultante, che chiameremo V-L JEPA per comodità, si articola in tre componenti principali.

Il primo è l'*encoder visivo* f_v: $\mathcal{O}$_v → $\mathbb{R}^d$, un ViT che mappa le patch dell'immagine o i tube video in rappresentazioni latenti. Il secondo è l'*encoder testuale* f_t: $\mathcal{O}$_t → $\mathbb{R}^d$, un Transformer che mappa sequenze di token linguistici in rappresentazioni latenti. Il terzo è il *predittore cross-modale* h_{v→t}: $\mathbb{R}^d$ × $\mathbb{R}^d$_pos → $\mathbb{R}^d$ che, data la rappresentazione visiva del contesto e un set di posizioni linguistiche target, predice le rappresentazioni linguistiche corrispondenti; e il suo duale h_{t→v}: $\mathbb{R}^d$ × $\mathbb{R}^d$_pos → $\mathbb{R}^d$ che predice rappresentazioni visive da contesto linguistico.

La loss della V-L JEPA è la somma di quattro termini corrispondenti ai quattro task di previsione possibili tra le due modalità:

$\mathcal{L}$_VL-JEPA = $\mathcal{L}$_{v→v}(masking visivo) + $\mathcal{L}$_{t→t}(masking linguistico) + $\mathcal{L}$_{v→t}(cross-modale visivo→testuale) + $\mathcal{L}$_{t→v}(cross-modale testuale→visivo)

con i termini intra-modali analoghi a I-JEPA e V-JEPA, e i termini cross-modali che richiedono di predire rappresentazioni di una modalità a partire dall'altra. Il termine $\mathcal{L}$_{v→t}, ad esempio, chiede al predittore di stimare la rappresentazione latente di porzioni di testo (tokens mascherati della didascalia) a partire dalla rappresentazione visiva dell'immagine corrispondente — una forma di captioning predittivo nello spazio latente. Formalmente:

$\mathcal{L}_{v→t}$ = $(1/|M_t|)$ $\Sigma_{p \in M_t}$ $\|\hat{h}_{v→t}(s_v, pos_p) / \|\hat{h}_{v→t}(s_v, pos_p)\| - s_t^{(p)} / \|s_t^{(p)}\|\|_2^2$

dove M_t è l'insieme dei token testuali mascherati, s_v è la rappresentazione della scena visiva, pos_p è la posizione del token mascherato, e s_t^{(p)} è la sua rappresentazione target (prodotta dall'encoder testuale EMA sul testo non mascherato). La normalizzazione L_2 previene il collasso e produce una loss invariante alla scala delle rappresentazioni.

Il vantaggio cruciale della V-L JEPA rispetto all'allineamento contrastivo CLIP è la struttura gerarchica del segnale di apprendimento. Mentre CLIP apprende l'allineamento a livello di intera immagine vs. intera didascalia, la V-L JEPA apprende l'allineamento a livello di patch visive locali vs. token linguistici locali, catturando le corrispondenze fine-grained tra regioni dell'immagine e referenti linguistici. Questa struttura locale è essenziale per il grounding nel senso di Harnad: non basta che "il gatto" corrisponda all'immagine del gatto globalmente, ma serve che le feature della regione dell'immagine contenente il gatto corrispondano alle feature del token linguistico "gatto".

6. Grounding spaziale e temporale: connessione con i world model embodied

Il grounding visio-linguistico nei world model embodied richiede non solo la capacità di mappare concetti linguistici a regioni visive statiche, ma anche la capacità di mappare verbi e predicati d'azione a trasformazioni temporali delle rappresentazioni visive — la *comprensione dell'azione* nel senso dell'intelligenza artificiale. Un robot che riceve l'istruzione "prendi il bicchiere e mettilo sul tavolo" deve essere capace di identificare il bicchiere nella scena (grounding nominale), di interpretare "prendi" come la sequenza di trasformazioni fisiche che portano dalla configurazione corrente alla configurazione con il bicchiere afferrato (grounding verbale-cinematico), e di interpretare "mettilo sul tavolo" come la sequenza di trasformazioni successive fino alla configurazione target (grounding spaziale-relazionale).

La formalizzazione del grounding verbale-cinematico può essere espressa come un mapping:

$\mathcal{M}$_action: $\mathbb{R}^d$_text × $\mathbb{R}^d$_state → $\mathbb{R}^d$_state_delta

dove $\mathbb{R}^d$_text è lo spazio delle rappresentazioni testuali dei verbi d'azione, $\mathbb{R}^d$_state è lo spazio degli stati latenti del world model, e $\mathbb{R}^d$_state_delta è lo spazio dei cambiamenti di stato prodotti dall'esecuzione dell'azione denotata dal verbo. Un predicato d'azione come "prendi" non corrisponde a un singolo delta di stato ma a una distribuzione di delta di stato — la distribuzione delle trasformazioni fisiche compatibili con la descrizione dell'azione nel contesto corrente.

Il framework CLIP-for-embodiment, proposto da Shridhar e collaboratori (2022) nell'architettura CLIPort, realizza questo mapping attraverso l'uso delle rappresentazioni CLIP come condizionamento per un transformer di trasformazione che genera le traiettorie di azione del robot. Le rappresentazioni CLIP dell'istruzione linguistica condizionano un *transport network* che predice le coppie (posizione di pick, posizione di place) dalle osservazioni visive — un grounding nominale e spaziale senza grounding verbale completo. Le architetture più recenti come SayCan (discussa nel capitolo sui modelli di linguaggio) estendono questo framework includendo il grounding verbale attraverso la affordance function.

La connessione con la JEPA gerarchica di LeCun è diretta: il livello inferiore della gerarchia JEPA corrisponde al grounding spaziale locale (regioni visive → parole), il livello intermedio al grounding d'azione (sequenze di regioni visive → verbi d'azione), e il livello superiore al grounding di obiettivi (stati finali desiderati → frasi di obiettivo). Una gerarchia di V-L JEPA annidate, ciascuna operante a una diversa scala temporale e spaziale, potrebbe implementare questo grounding a livelli multipli in modo end-to-end — apprendendo le corrispondenze tra linguaggio e percezione all'interno dello stesso framework di predizione latente che è già dimostrato efficace per l'apprendimento delle rappresentazioni visive.

7. FLAVA, CoCa e unified architectures: verso encoder universali

La ricerca sui modelli visio-linguistici fondazionali si è biforcata in due approcci architetturali distinti: i modelli *dual encoder* (come CLIP) che usano encoder separati per le due modalità e li allineano contrastivamente, e i modelli *fusion encoder* (come ViLBERT, UNITER, OSCAR) che usano meccanismi di fusione cross-modale (tipicamente cross-attention) per costruire rappresentazioni congiunte dall'inizio. I modelli duali sono computazionalmente efficienti per il retrieval (le rappresentazioni possono essere pre-calcolate e indicizzate) ma limitati nella granularità del grounding; i modelli fusion permettono grounding fine-grained ma richiedono l'elaborazione congiunta di ogni coppia (immagine, testo) a inference time.

FLAVA (Singh e collaboratori 2022, Meta) unifica i due approcci in un'unica architettura che combina tre modalità di training: unimodale visivo (un ViT addestrato con MAE), unimodale testuale (un Transformer addestrato con MLM, Masked Language Modeling), e multimodale (fusion di entrambe le modalità attraverso un encoder di fusione con cross-attention, addestrato con una combinazione di loss contrastiva, ITM e MLM multimodale). La loss totale di FLAVA è:

$$\mathcal{L}_FLAVA = \mathcal{L}_unimodal_v + \mathcal{L}_unimodal_t + \lambda_1 \mathcal{L}_contrastive + \lambda_2 \mathcal{L}_ITM + \lambda_3 \mathcal{L}_MLM_multimodal + \lambda_4 \mathcal{L}_MAE$$

dove $\mathcal{L}$_ITM è la Image-Text Matching loss (classificazione binaria della corrispondenza coppia immagine-testo), $\mathcal{L}$_MLM_multimodal è la loss di masked language modeling con contesto visivo (il modello deve predire i token testuali mascherati avendo accesso alle feature visive), e $\mathcal{L}$_MAE è la loss di masked autoencoding visiva. Il training congiunto su tutti questi obiettivi produce un modello che eccelle sia in task unimodali che multimodali — al costo di una complessità di training superiore.

CoCa (Contrastive Captioners, Yu e collaboratori 2022, Google) introduce una divisione architetturale elegante: un *unimodal image encoder* e un *multimodal text decoder* separati, dove il decoder usa cross-attention sulle feature dell'image encoder. La loss combina la perdita contrastiva (applicata al pooling globale delle feature) e la perdita di captioning autoregressiva (applicata al decoder cross-attentivo):

$$\mathcal{L}_CoCa = \mathcal{L}_contrastive(pool(f_v(I)), pool(f_t(T))) + \mathcal{L}_captioning(p_\theta(T \mid f_v(I)))$$

La separazione tra image encoder (ottimizzato per la qualità della rappresentazione globale attraverso la loss contrastiva) e text decoder (ottimizzato per il grounding fine-grained attraverso la loss di captioning) permette di scalare i due componenti indipendentemente e di usarli separatamente per task che ne richiedono solo uno.

Per i world model, CoCa offre una struttura particolarmente adatta: l'image encoder fornisce la rappresentazione dello stato visivo, il text decoder genera la descrizione linguistica dello stato (utile per il ragionamento linguistico sull'obiettivo e per l'interfaccia con l'utente), e la loss contrastiva garantisce che le rappresentazioni globali siano allineate per il retrieval e la classificazione zero-shot. L'integrazione di un CoCa pre-addestrato come backbone visio-linguistico di un world model RSSM è

l'architettura sottostante a diversi sistemi robotici recenti (incluso RT-2 nella sua variante con VLM fondazionale).

8. Structured World Models con grounding linguistico: riferimento, predicazione e co-composizionalità

Il grounding linguistico nei world model non si limita al mapping tra concetti semplici e regioni visive, ma include strutture semantiche complesse come la predicazione (il soggetto possiede una proprietà), la relazione spaziale (X è sopra Y), la quantificazione (tutti i cubi rossi), e la composizione di queste strutture in descrizioni complesse. La capacità di grounding su queste strutture semantiche complesse è il test discriminante tra un sistema che "comprende" il linguaggio nel senso funzionalista e uno che memorizza pattern statistici superficiali.

La compositional generalization — la capacità di generalizzare a nuove combinazioni di concetti visti separatamente — è il benchmark metodologicamente più rigoroso per la comprensione genuina del linguaggio. SCAN (Lake e Baroni 2018), COGS (Kim e Linzen 2020) e COMPGUESS (Nie e collaboratori 2020) sono dataset che testano specificatamente questa capacità: un sistema che generalizza compositivamente deve essere capace di eseguire correttamente "cammina due volte e salta" anche se "cammina" e "salta" sono stati visti solo separatamente, e "due volte" solo con altre azioni.

Il mapping tra questa struttura semantica compositiva e le trasformazioni dello stato visivo nei world model richiede architetture capaci di decomposizione strutturata. Le *neuro-symbolic world models* — che combinano un neural backbone per la percezione con un reasoner simbolico per il piano e la composizione — sono la risposta più diretta a questa sfida. NS-VQA (Neural-Symbolic Visual Question Answering, Yi e collaboratori 2018) e CLEVRER (Collision Events for Video Representation and Reasoning, Yi e collaboratori 2020) mostrano come separare la percezione (una rete neurale che estrae le proprietà degli oggetti) dal ragionamento (un programma simbolico che combina le proprietà attraverso operatori logici) permetta di ottenere grounding compositivo quasi perfetto su scene 3D sintetiche.

La sfida per i world model realistici è che le scene del mondo fisico non si prestano a una decomposizione simbolica esatta come le scene sintetiche di CLEVRER: gli oggetti sono parzialmente occlusi, le loro proprietà cambiano continuamente, e le relazioni tra di loro sono spesso graduali invece che booleane. Le architetture neuro-symbolic soft — che usano il linguaggio come struttura di programmazione ma implementano le operazioni attraverso funzioni neurali continue — rappresentano un compromesso: le operazioni logiche composizionali sono implementate come combinazioni differenziabili delle rappresentazioni neurali, preservando la compositività strutturale senza richiedere la discretizzazione simbolica completa.

9. CLIP come funzione di ricompensa per il reinforcement learning robotico

Una delle applicazioni più immediate e praticamente rilevanti dei modelli visio-linguistici per i world model riguarda l'uso di CLIP come funzione di ricompensa per il reinforcement learning. La definizione di funzioni di ricompensa per task robotici complessi è storicamente uno dei problemi più difficili della robotica — il *reward engineering* richiede una precisa specificazione matematica dell'obiettivo desiderato, spesso difficile da tradurre in forma algoritmica. Le descrizioni in linguaggio naturale degli obiettivi — "metti il cub rosso nella scatola verde" — sono molto più naturali da specificare per gli esseri umani, e CLIP permette di tradurre queste descrizioni in segnali di ricompensa.

Il reward CLIP è definito come la similarità coseno tra la rappresentazione CLIP dell'osservazione corrente e la rappresentazione CLIP della descrizione linguistica dell'obiettivo:

$$r_{CLIP}(o_t; g) = \cos(f_v(o_t), f_t(g)) = f_v(o_t)^\top f_t(g) / (\|f_v(o_t)\| \, \|f_t(g)\|)$$

dove o_t è l'osservazione visiva al passo t e g è la descrizione linguistica dell'obiettivo. Questo reward è massimizzato quando l'osservazione visiva è semanticamente simile alla descrizione dell'obiettivo nello spazio di embedding CLIP — una condizione che approssima la realizzazione dell'obiettivo descritto.

CLIP-Reward (Rocamonde e collaboratori 2023) e R3M (Nair e collaboratori 2022) mostrano che questo reward è efficace per guidare il reinforcement learning su task robotici di manipolazione, specialmente in combinazione con un world model che permette di pianificare con il reward CLIP in ambienti simulati. Il vantaggio è la specificazione zero-shot degli obiettivi — nessun esempio dimostrativo o funzione di ricompensa ad-hoc è necessario, solo la descrizione linguistica dell'obiettivo desiderato.

Le limitazioni del reward CLIP come segnale di rinforzo emergono nelle situazioni dove le similarità di embedding CLIP non catturano le proprietà fisiche rilevanti. CLIP è addestrato su coppie (immagine, didascalia) dove la didascalia descrive il contenuto dell'immagine — non le sue proprietà fisiche sottostanti. Pertanto, la similarità CLIP può essere alta per immagini visivamente simili che differiscono nelle proprietà fisicamente critiche (ad esempio, due immagini che mostrano un oggetto nella stessa posizione ma con orientazioni diverse che sono invisibili nella proiezione 2D). Il reward CLIP può quindi produrre falsi positivi — stati che sembrano raggiungere l'obiettivo secondo CLIP ma non lo raggiungono fisicamente.

La soluzione proposta da Hulse e collaboratori (2024) è il *grounded reward shaping*: il reward CLIP viene combinato con un reward di grounding 3D che misura la realizzazione fisica dell'obiettivo attraverso un modello di stima della posa degli oggetti (come FoundationPose o BundleSDF). La funzione di ricompensa composita:

$$r_{total}(o_t; g) = \lambda_{CLIP} \cdot r_{CLIP}(o_t; g) + \lambda_{3D} \cdot r_{3D}(o_t; g)$$

combina la facilità di specificazione del reward linguistico con la precisione fisica del reward geometrico, producendo un segnale di rinforzo che è sia facilmente specificabile dagli utenti sia fisicamente accurato.

10. DALL-E, Stable Diffusion e modelli di generazione condizionata come world model visivi

I modelli di generazione di immagini condizionata al testo — DALL-E (Ramesh e collaboratori 2021, 2022), Stable Diffusion (Rombach e collaboratori 2022), Imagen (Saharia e collaboratori 2022) — realizzano un mapping direzionale dal testo all'immagine: data una descrizione linguistica, generano un'immagine che la soddisfa visivamente. Dal punto di vista dei world model, questo mapping è l'opposto del reward CLIP: invece di mappare immagini a concetti, mappa concetti a immagini — implementando una forma di *immaginazione guidata dal linguaggio*.

La struttura di Stable Diffusion è particolarmente rilevante per i world model. Il modello esegue la diffusione nello spazio latente di un VQVAE (dimensione tipicamente 4×H/8×W/8 invece di 3×H×W), condizionando il denoiser U-Net sulle rappresentazioni CLIP del prompt testuale attraverso cross-attention. L'operazione di condizionamento è:

U-Net_θ(x_t, t, c_text) = U-Net_θ(x_t, t) + Σ_l CrossAttn(Q_l(x_t), K_l(c_text), V_l(c_text))

dove la cross-attention a ogni livello l dell'U-Net proietta le feature latenti (query) sulle rappresentazioni testuali (chiavi e valori), permettendo al processo di denoising di incorporare selettivamente le informazioni linguistiche più rilevanti per la generazione a ogni scala spaziale. Le scale fini della U-Net (basso livello) catturano dettagli visivi locali; le scale grossolane (alto livello) catturano la struttura semantica globale.

Per i world model, Stable Diffusion come modello di transizione condizionato sull'azione è esattamente il paradigma di UniSim discusso nel capitolo sulle leggi di scala: dato lo stato visivo corrente o_t e l'azione a_t (codificata come testo, come vettore continuo, o come embedding), il modello di diffusione genera il successivo stato visivo o_{t+1} plausibile. Il classifier-free guidance (CFG, Ho e Salimans 2021) permette di controllare il trade-off tra fedeltà alla condizione linguistica e diversità della generazione:

ε_θ^{guided}(x_t, t, c) = ε_θ(x_t, t, ∅) + w [ε_θ(x_t, t, c) - ε_θ(x_t, t, ∅)]

dove w > 1 è il guidance weight, ε_θ(x_t, t, ∅) è il modello non condizionato (con condizione droppa a ∅ durante il training con probabilità p_drop), e ε_θ(x_t, t, c) è il modello condizionato. Per w > 1, la generazione è più fedele alla condizione c ma meno diversa; per w = 1, coincide con la generazione condizionata standard; per w → ∞, il modello genera il campione che massimizza la log-verosimiglianza condizionale.

La connessione con la JEPA è manifesta nel confronto tra i due approcci al condizionamento: JEPA condiziona la previsione sullo spazio latente (non sullo spazio dei pixel), e il predittore opera sulle rappresentazioni semantiche invece che sul segnale grezzo. Stable Diffusion come world model condiziona la generazione sullo spazio dei pixel, ma il conditioning testuale attraverso CLIP fornisce un ancora semantica che orienta la generazione. La V-L JEPA con decoder di diffusione realizza la sintesi: condizionamento semantico nello spazio latente, con la possibilità di materializzare le previsioni in immagini attraverso il decoder di diffusione quando necessario — un'architettura che combina l'efficienza predicittiva della JEPA con la ricchezza visiva della diffusione.

11. Valutazione del grounding visio-linguistico: RefCOCO, VQA e benchmark composizionali

La valutazione del grounding visio-linguistico nei world model richiede benchmark che vadano oltre la classificazione zero-shot di CLIP e misurino la qualità del mapping semantico a livello locale e composizionale. I benchmark standard per questo scopo includono RefCOCO, VQA, e le loro varianti composizionali.

RefCOCO (Kazemzadeh e collaboratori 2014) richiede di identificare la regione dell'immagine referenziata da una frase nominale in linguaggio naturale — "il cane più grande", "la terza sedia da sinistra", "l'uomo con la giacca rossa". La valutazione è in termini di Intersection over Union (IoU) tra il bounding box predetto e quello annotato:

$$\text{IoU}(B_{pred}, B_{gt}) = |B_{pred} \cap B_{gt}| \,/\, |B_{pred} \cup B_{gt}|$$

con accuratezza computata come frequenza di IoU > 0.5. I modelli allo stato dell'arte (InternVL-2, GPT-4V) raggiungono accuratezze di ~90% su RefCOCO standard, ma degradano a ~70% sulle varianti composizionali che richiedono di referenziare entità definite da relazioni spaziali complesse tra più oggetti.

VQA (Visual Question Answering, Antol e collaboratori 2015) richiede di rispondere a domande in linguaggio naturale sull'immagine. Le domande variano da attributi semplici ("di che colore è il cappello?") a ragionamento spaziale ("c'è qualcosa tra il banco e la finestra?") a conteggio ("quante persone ci sono?"). L'accuratezza standard su VQAv2 per i modelli migliori supera l'85% — ma le domande che richiedono ragionamento compositivo o causale ("se rimuovessi il vaso cosa restereste a guardare?") rimangono sfidanti con accuratezze di ~60-70%.

Il benchmark Winoground (Thrush e collaboratori 2022) è particolarmente discriminativo per il grounding compositivo: ogni esempio consiste di due immagini e due didascalie dove le stesse parole sono composte diversamente — "il cane è sopra il gatto" vs. "il gatto è sopra il cane". Un modello con vero grounding compositivo deve essere capace di distinguere correttamente quale immagine corrisponde a quale didascalia, cosa che CLIP e modelli simili falliscono sistematicamente (~20% di

accuratezza su Winoground per i modelli contrastivi, poco sopra il caso). Il fallimento di CLIP su Winoground rivela che l'allineamento contrastivo su paia globali non è sufficiente per acquisire il grounding compositivo — confermando la necessità di architetture come la V-L JEPA con allineamento locale.

12. Prospettive: verso world model con comprensione multimodale profonda

La traiettoria di ricerca delineata in questo capitolo converge verso una visione di world model che integrano comprensione visiva, linguistica, e fisica in un unico framework coerente. La JEPA gerarchica di LeCun — con livelli di astrazione crescente che mappano le trasformazioni sensoriali a scale temporali sempre più lunghe — offre il framework architetturale per questa integrazione: ogni livello della gerarchia può specializzarsi su un tipo di invarianza (spaziale, temporale, causale, semantica), e il linguaggio può essere introdotto come modalità aggiuntiva che condivide gli stessi meccanismi di predizione latente.

Il paradigma della *grounded compositional world model* emerge come l'obiettivo più ambizioso: un sistema che apprende rappresentazioni degli stati del mondo che sono simultaneamente percettivamente ricche (catturano le caratteristiche visive rilevanti), linguisticamente grounded (corrispondono ai concetti espressi nel linguaggio naturale), causalmente strutturate (rispettano le invarianze causali del mondo fisico), e composizionalmente generalizzabili (permettono di comprendere configurazioni nuove combinando concetti visti separatamente). Nessun sistema attuale soddisfa tutti questi requisiti simultaneamente; i sistemi più avanzati — GPT-4V, Gemini Ultra, Claude 3 Opus — li soddisfano parzialmente ma mostrano sistematici fallimenti in ciascuna dimensione.

Il percorso verso questa comprensione integrata passa inevitabilmente attraverso il tipo di esperienza sensori-motoria che gli esseri umani acquisiscono fin dall'infanzia — l'interazione fisica con gli oggetti, il feedback tattile, la coordinazione occhio-mano, l'apprendimento del linguaggio in un contesto di manipolazione e comunicazione incarnate. La robotica incarnata non è quindi un'applicazione dei world model: è la condizione necessaria per lo sviluppo di world model che possiedano genuina comprensione del mondo fisico e sociale. Questa intuizione — formulata da Piaget, ripresa da Varela e dalla cognitive science embodied, e oggi riaffermata dall'esperienza empirica dell'AI — suggerisce che la prossima frontiera non sarà un modello più grande addestrato su più testo, ma un agente fisicamente situato che impara il mondo interagendo con esso, guidato da un'architettura di predizione latente gerarchica che integra modalità percettive multiple con il linguaggio naturale come interfaccia di comunicazione e di specificazione degli obiettivi.

XVII. Fisica statistica dell'apprendimento profondo e transizioni di fase nei world model

1. Il programma della meccanica statistica applicata alle reti neurali: origini e ambizioni

Il ricorso alla fisica statistica come strumento di analisi dei sistemi di apprendimento automatico non costituisce una novità episodica o una moda transitoria, bensì una linea di ricerca con radici profonde e una coerenza interna notevole. La ragione fondamentale di questo trasferimento concettuale risiede in una somiglianza strutturale profonda tra i sistemi di molte particelle in interazione, oggetto della termodinamica e della meccanica statistica classica, e le reti neurali profonde, dove un numero elevatissimo di parametri interagisce attraverso la funzione di perdita in modo da produrre comportamenti collettivi qualitativamente distinti dalla somma dei contributi individuali. Gardner e Derrida, alla fine degli anni Ottanta, furono tra i primi a riconoscere che la geometria dello spazio delle soluzioni di un problema di apprendimento — l'insieme dei vettori di parametri compatibili con un certo insieme di esempi — poteva essere studiata con le stesse tecniche sviluppate per i vetri di spin e i sistemi disordinati. Le transizioni di fase, i fenomeni critici e la comparsa di ordine a lunga distanza, che nella fisica dei materiali segnalano cambiamenti qualitativi nelle proprietà macroscopiche, trovano corrispondenza precisa nel comportamento delle reti neurali al variare della capacità, della quantità di dati, e della struttura dell'ottimizzazione.

Per i modelli del mondo, questo programma è di rilevanza speciale per almeno tre ragioni distinte. La prima è che i world model sono sistemi ad altissima dimensionalità — miliardi di parametri che modellano distribuzioni su spazi di stati comparabilmente grandi — dove i fenomeni collettivi e le transizioni di fase sono attesi a priori, non come eccezioni ma come regola. La seconda è che le transizioni di fase nel comportamento dei world model durante il training — il passaggio dal regime di correlazione debole al regime di rappresentazione strutturata, l'emergenza improvvisa della capacità predittiva — sono fenomeni empiricamente osservati ma teoricamente poco compresi, e la fisica statistica offre gli strumenti formali per caratterizzarli rigorosamente. La terza, forse la più profonda, è che la distinzione tra regime di kernel e regime di feature learning — discussa nel capitolo sull'ottimizzazione in relazione al NTK — è formalmente analoga alla distinzione tra la fase ferromagnetica e la fase disordinata nei sistemi di spin: una transizione di fase, mediata dalla forza delle interazioni (la learning rate) e dalla struttura delle correlazioni nei dati.

2. Reti neurali come sistemi disordinati: la connessione con i vetri di spin

Il modello di Sherrington-Kirkpatrick (SK) è il prototipo dei vetri di spin con interazioni a lungo raggio: N spin $\sigma_i \in \{\pm 1\}$ che interagiscono a coppie attraverso accoppiamenti J_{ij} estratti da una distribuzione gaussiana con media zero e varianza J^2/N. La funzione hamiltoniana è:

$\mathcal{H}_{SK}(\{\sigma\}) = -\sum_{i<j} J_{ij} \sigma_i \sigma$

Il fattore 1/N nella varianza degli accoppiamenti garantisce che l'energia sia estensiva. Il sistema mostra una transizione di fase a temperatura critica T_c = J dal paramagnetico disordinato alla fase vetro di spin, caratterizzata dalla comparsa di una magnetizzazione locale non nulla ma con orientazione casuale — gli spin si ↑congelano↑ in configurazioni quasi-statiche che variano da campione a campione del disordine.

La connessione con le reti neurali è multipla. Una rete di Hopfield (discussa nel capitolo sulla memoria) con N neuroni e P pattern memorizzati $\{\xi^m\}_{=1}^P$ ha una funzione di energia equivalente a quella di SK quando P cresce come P = αN con $\alpha \in (0, 1)$. Per $\alpha < \alpha_c \approx 0.138$, la rete recupera i pattern memorizzati correttamente (fase ferromagnetica); per $\alpha > \alpha_c$, i pattern subiscono interferenza reciproca e la rete entra in una fase vetro di spin dove gli attrattori sono configurazioni spurie non correlate con nessun pattern singolo. Questa transizione — che nella fisica statistica dei vetri è caratterizzata dalla comparsa di una funzione di overlap q(τ) non banale nello spazio delle repliche — corrisponde, nei modelli del mondo, al passaggio da un regime dove il modello apprende rappresentazioni coerenti a uno dove le rappresentazioni codificano correlazioni spurie tra i dati di training.

Il metodo delle repliche, sviluppato da Parisi e collaboratori negli anni Settanta-Ottanta, calcola la funzione di partizione libera media $\langle \log Z \rangle_J$ attraverso l'identità delle repliche:

$$\langle \log Z \rangle_J = \lim_{n \to 0} \left(\langle Z^n \rangle_J - 1 \right) / n$$

La media su n repliche del sistema introduce correlazioni tra le repliche che, nella soluzione di Parisi, sono descritte da una funzione di overlap q(x) per $x \in [0, 1]$ — la parametrizzazione della rottura di simmetria tra le repliche. Nella fase vetro di spin, q(x) è una funzione non costante (rottura di simmetria delle repliche completa, o RSB), indicando che lo spazio delle soluzioni si frammenta in un'infinità di bacini di attrazione gerarchicamente organizzati, con barriere di energia che crescono con le dimensioni del sistema.

Per le reti neurali profonde, l'analogo della RSB è la struttura ultrametrica dello spazio dei minimi della funzione di perdita: due minimi arbitrari hanno una distanza in spazio dei parametri che è determinata dalla loro appartenenza a diversi bacini nell'albero gerarchico delle soluzioni. Questa struttura ultrametrica — che nel vetro di spin di Parisi è rigorosa nel limite termodinamico $N \to \infty$ — è stata osservata empiricamente nelle reti neurali profonde da Chaudhari e collaboratori (2017) attraverso l'analisi delle connessioni tra i minimi nell'albero delle distanze in spazio dei parametri.

3. La distribuzione degli autovalori dell'Hessiano: la matrice casuale di Wigner e i limiti di Marchenko-Pastur

Un approccio sistematico alla struttura locale del paesaggio di perdita dei world model richiede l'analisi della distribuzione degli autovalori dell'Hessiano $H(\theta^*)$ in un minimo θ^*. Per reti neurali molto profonde con milioni di parametri, questa matrice è di dimensione $n \times n$ con $n \sim 10^7 - 10^9$ — intrattabile nella sua completezza. La teoria delle matrici casuali (Random Matrix Theory, RMT) fornisce risultati analitici precisi sulla distribuzione limite degli autovalori di matrici con specifiche strutture di correlazione, applicabili a questa situazione come limite termodinamico.

La legge di Marchenko-Pastur descrive la distribuzione degli autovalori della matrice di covarianza campionaria $X\,X^T / N$ dove $X \in \mathbb{R}^{mxN}$ è una matrice con elementi i.i.d. gaussiani con varianza σ^2. Nel limite $m, N \to \infty$ con rapporto fisso $\gamma = m/N$, la densità empirica degli autovalori converge quasi certamente alla distribuzione di Marchenko-Pastur:

$$\varrho_MP(\lambda) = (1 / 2\pi\,\sigma^2\,\gamma\,\lambda)\,\sqrt{((\lambda_+ - \lambda)(\lambda - \lambda_-))} \text{ per } \lambda \in [\lambda_-, \lambda_+]$$

con $\lambda\pm = \sigma^2\,(1 \pm \sqrt{\gamma})^2$. Per $\gamma < 1$ ($m < N$, sistema sottodeterminato), la distribuzione ha un atomo di massa $(1 - \gamma)$ in $\lambda = 0$ corrispondente agli autovalori nulli. Per $\gamma > 1$ ($m > N$, sistema sovradeterminato), tutti gli autovalori sono positivi.

Martin e Mahoney (2018, 2019, 2021) hanno mostrato empiricamente che la distribuzione degli autovalori delle matrici di peso dei layer di reti neurali profonde ben addestrate non segue la legge di Marchenko-Pastur, bensì una distribuzione a coda pesante con comportamento asintotico:

$$\varrho(\lambda) \sim \lambda\^\{-\alpha-1\} \text{ per } \lambda \to \infty$$

con esponente $\alpha \in (2, 4)$ per reti ben generalizzanti. Questo comportamento a coda pesante — indicativo di una struttura correlazionale a lungo raggio tra i pesi — è assente nelle reti non addestrate o in quelle overfit, e costituisce una firma della qualità della generalizzazione. La connessione con la fisica statistica è profonda: la distribuzione a coda pesante è caratteristica della criticità — il comportamento critico dei sistemi fisici al punto di transizione di fase è descritto da leggi di potenza con esponenti universali che non dipendono dai dettagli microscopici del sistema. Le reti che generalizzano bene operano, in questo senso, ↑vicino alla criticità↑ nel loro spazio dei pesi.

Per i modelli del mondo di tipo RSSM, la struttura della matrice di peso del modello di transizione f_φ — che è la componente più critica per la qualità delle previsioni — dovrebbe mostrare questa firma di criticità se il modello ha acquisito strutture dinamiche generalizzabili. Martin e Mahoney propongono di usare l'esponente α come indicatore di salute del modello senza necessità di test su dataset separati — una metrica di qualità puramente interna all'architettura, derivabile dall'analisi spettrale dei pesi appresi.

4. La transizione di fase nell'apprendimento: il framework della complessità di campionamento

Il problema fondamentale della teoria dell'apprendimento è determinare quanti esempi siano necessari per apprendere una funzione target con accuratezza ε e confidenza δ. La teoria PAC (Probably Approximately Correct) di Valiant fornisce un limite superiore alla complessità di campionamento per classi di ipotesi finite:

$m \geq (1/\varepsilon) (\log |\mathcal{H}| + \log(1/\delta))$

dove $|\mathcal{H}|$ è la cardinalità della classe di ipotesi. Per classi infinite, la dimensione di Vapnik-Chervonenkis (VC) $d_{VC}(\mathcal{H})$ sostituisce $\log |\mathcal{H}|$, e i limiti diventano:

$m \geq C \cdot (d_{VC} / \varepsilon) \cdot \log(1/\varepsilon \delta)$

per costanti C appropriate. Tuttavia, questi limiti sono generalmente molto pessimistici per le reti neurali profonde, che mostrano una complessità di campionamento molto inferiore a quanto predetto dalla dimensione VC — a volte inferiore di ordini di grandezza.

La fisica statistica offre un'alternativa: calcolare la complessità di campionamento non attraverso la dimensione VC ma attraverso l'analisi termodinamica dello spazio delle soluzioni. Si definisce il volume dello spazio delle soluzioni V(m) come la misura del sottoinsieme di $\mathcal{H}$ (nello spazio dei parametri) che classifica correttamente m esempi casuali. La relazione tra V(m) e m mostra tipicamente una transizione di fase sharpa intorno a m_c:

per $m << m_c$: $V(m) \approx |\mathcal{H}|$ (quasi tutte le ipotesi sono compatibili) per $m >> m_c$: $V(m) \to 0$ esponenzialmente in m

Il punto critico m_c è analogo alla temperatura critica T_c dei sistemi fisici e corrisponde alla quantità di dati sufficiente per ridurre il volume delle soluzioni a una singola ipotesi (o a un insieme molto piccolo). Per le reti neurali, questa transizione è smussata dalla alta dimensionalità e dalla distribuzione delle soluzioni nello spazio dei parametri, ma il comportamento qualitativo rimane: sotto la soglia critica, molte reti diverse si addestrano ugualmente bene (alto volume nello spazio delle soluzioni); sopra la soglia, la rete converge su una struttura tipica relativamente stabile.

Per i world model, la transizione di complessità di campionamento ha un'interpretazione concreta: m_c è il numero minimo di traiettorie di interazione necessario perché il modello del mondo ↑emerga↑ — nel senso che le sue previsioni diventano significativamente più accurate di una previsione di riferimento (media empirica). Nella pratica, DreamerV3 e architetture comparabili mostrano curve di apprendimento con una caratteristica forma a ↑ginocchio↑: l'accuratezza rimane bassa per un numero di passi di training proporzionale al numero di parametri del modello, poi cresce bruscamente — suggerendo una transizione di fase nell'apprendimento piuttosto che un progresso graduale.

5. Double descent e la curva di generalizzazione non monotona

Il fenomeno del double descent, documentato sistematicamente da Belkin e collaboratori (2019) e da Nakkiran e collaboratori (2020), rivela che la curva dell'errore di test di una rete neurale in funzione del numero di parametri o del tempo di training non è monotonamente crescente o decrescente, bensì presenta una caratteristica forma a doppia decrescita separata da un picco critico.

Formalmente, si definisce un regime ↑interpolation threshold↑: per n parametri e m esempi di training, il regime sottodeterminato corrisponde a $n < m$ (la rete non ha sufficiente capacità per memorizzare il dataset) e il regime sovradeterminato a $n > m$. Al punto $n = m$ si trova il picco critico dell'errore di test — la rete ha esattamente abbastanza parametri per memorizzare perfettamente il training set, ma solo attraverso soluzioni altamente discontinue che non generalizzano. Per $n > m$ (regime fortemente sovradeterminato), l'errore di test comincia a decrescere di nuovo — il regime ben documentato empiricamente dove reti molto grandi generalizzano sorprendentemente bene.

La spiegazione fisica del double descent si trova nell'analisi delle singole soluzioni raggiunte dalla discesa del gradiente. Al picco critico ($n \approx m$), la norma minima della soluzione interpolante — il minimo della norma $\|\theta\|_2$ tra tutti i θ che annullano la loss di training — è molto grande: la soluzione è una perturbazione di alta ampiezza del punto iniziale nello spazio dei parametri. Per $n >> m$, la soluzione di norma minima ha $\|\theta\|_2 \sim \sqrt{(m/n)} \to 0$ per $n \to \infty$, producendo una perturbazione di bassa ampiezza che corrisponde a una funzione altamente regolarizzata. Il double descent è quindi l'effetto dell'ottimizzazione implicita della discesa del gradiente: tra tutte le soluzioni che interpolano il training set, la discesa del gradiente converge alla soluzione di norma minima (nel limite del gradiente flusso), e questa norma ha una dipendenza non monotona dal numero di parametri.

Bartlett e collaboratori (2020) derivano un limite superiore all'errore di test per il regressore lineare di norma minima che esplicita questa dipendenza. Per dati gaussiani in $\mathbb{R}^d$ con $d >> n$, il classificatore di norma minima ha errore di classificazione:

$$E[\text{err}] \leq \|\beta^*\|_2^2 \, \sigma^2 \, (n/(d-n)) + \varepsilon_{\text{noise}}$$

dove β^* è il vero parametro e σ^2 è la varianza del rumore. Il termine $n/(d-n)$ cresce monotonamente verso 1 per $n \to d$ (peak di interpolazione) e poi decresce per $d \to \infty$, spiegando la forma a doppia discesa. Nel regime fortemente sovradeterminato $n << d$, l'errore di test è proporzionale a n/d — e si riduce tanto più quanto più è grande la dimensione dello spazio (effetto di benign overfitting).

Per i world model, il double descent ha un'implicazione pratica: aggiungere parametri al modello di transizione non necessariamente migliora le previsioni se ci si trova in prossimità del picco critico — potrebbe peggiorarle temporaneamente prima che l'ulteriore aggiunta di capacità inizi a portare benefici nel regime fortemente sovradeterminato. Questo spiega perché i world model di piccola dimensione talvolta superano quelli di media dimensione su dataset di training di dimensione fissa, fenomeno che appare paradossale se interpretato con la classica visione del trade-off bias-varianza.

6. Termodinamica dell'apprendimento: temperatura, entropia e capacità di memorizzazione

La distribuzione di Gibbs-Boltzmann applicata alla funzione di perdita di una rete neurale definisce un ensemble statistico:

$$p_\beta(\theta) = \exp(-\beta\, L(\theta)) \,/\, Z(\beta)$$

dove $\beta = 1/T$ è la temperatura inversa e $Z(\beta) = \int \exp(-\beta\, L(\theta))\, d\theta$ è la funzione di partizione. Questa distribuzione assegna alta probabilità ai parametri con loss bassa (bassa energia), e la temperatura T controlla il trade-off tra esplorazione e sfruttamento: T grande corrisponde a una distribuzione quasi uniforme (esplorazione), T piccola concentra la probabilità intorno ai minimi della loss (sfruttamento).

L'entropia della distribuzione di Gibbs è:

$$S(T) = -\int p_\beta(\theta) \log p_\beta(\theta)\, d\theta = \beta\, \langle L \rangle_\beta + \log Z(\beta)$$

dove $\langle L \rangle_\beta$ è la loss media alla temperatura β. La capacità calorica del sistema — l'analogo fisico della suscettibilità — è:

$$C(T) = \beta^2\, \partial \langle L \rangle_\beta / \partial \beta = \beta^2\, \mathrm{Var}_\beta[L(\theta)]$$

La capacità calorica misura le fluttuazioni della loss alla temperatura T: picchi in C(T) segnalano transizioni di fase nel paesaggio di perdita. Praticamente, la capacità calorica può essere stimata dalle fluttuazioni della loss durante il training SGD: il batch loss ha varianza ~ $\mathrm{Var}_\beta[L(\theta)]$ se il minibatch è campionato uniformemente.

Chaudhari e collaboratori (2016) propongono il principio di *entropy-SGD*: invece di minimizzare la loss L(θ), minimizzare la ↑local entropy↑ — la log-probabilità di Gibbs in un intorno sferico di raggio γ attorno a θ:

$$F_\gamma(\theta) = -(1/\beta) \log \int_{\{\|\delta\| \le \gamma\}} \exp(-\beta\, L(\theta + \delta))\, d\delta$$

La local entropy $F_\gamma(\theta)$ è bassa nei minimi piatti dove $L(\theta + \delta)$ è bassa per tutte le perturbazioni $\|\delta\| \le \gamma$, e alta nei minimi acuti dove $L(\theta + \delta)$ cresce rapidamente. Minimizzare F_γ invece di L realizza quindi una ricerca di minimi piatti — lo stesso obiettivo della SAM discussa nel capitolo sull'ottimizzazione, ma derivata dal principio termodinamico della minimizzazione dell'energia libera invece che da argomenti di generalizzazione PAC-Bayes.

Il gradiente di F_γ rispetto a θ è:

$$\nabla_\theta F_\gamma(\theta) = \int (p_\gamma(\delta;\, \theta) \cdot \nabla_\theta L(\theta + \delta))\, d\delta$$

dove p_γ(δ; θ) ∝ exp(−β L(θ + δ)) 𝟙[‖δ‖ ≤ γ] è la distribuzione di Gibbs locale. Questo gradiente è una media pesata dei gradienti di L in un intorno sferico, dando peso maggiore alle direzioni dove L è più bassa (quelle che ↑tirano↑ verso il minimo piatto). In pratica, la local entropy viene approssimata attraverso Langevin dynamics breve per stimare ⟨∇*θ L(θ + δ)*⟩{p_γ}.

Per i world model, la minimizzazione della local entropy invece della loss standard dovrebbe produrre rappresentazioni latenti che rimangono stabili rispetto a piccole perturbazioni nei parametri — una proprietà direttamente collegata alla robustezza delle previsioni rispetto all'incertezza del modello. Un world model addestrato con entropy-SGD o SAM dovrebbe quindi produrre previsioni più coerenti anche in regioni dello spazio degli stati poco rappresentate nel dataset di training, migliorando la generalizzazione fuori distribuzione che è il limite principale dei world model per la robotica reale.

7. Il limite termodinamico delle reti neurali: il regime Mean Field

La teoria del campo medio (*Mean Field theory*, MFT) è la tecnica di approssimazione fondamentale della meccanica statistica per sistemi con interazioni a lungo raggio. L'idea è di sostituire le interazioni esatte tra ogni coppia di particelle con un campo medio — l'effetto medio di tutte le altre particelle su una singola — riducendo il problema a molti corpi a un problema a singola particella in un campo effettivo.

Per le reti neurali, la teoria del campo medio nel limite di ampiezza infinita (infinite-width limit) è stata sviluppata da Neal (1996) per i layer singoli e da Jacot, Gabriel e Hongler (2018) attraverso il NTK, già discusso nel capitolo sull'ottimizzazione. In questo limite, ogni neurone del layer l riceve input da un numero infinito di neuroni del layer l−1, e le fluttuazioni del suo pre-attivazione si cancellano per la legge dei grandi numeri, convergendo a una distribuzione gaussiana. Questo è il ↑Central Limit Theorem applicato alle reti neurali↑: per neuroni di larghezza m con parametri i.i.d. di varianza 1/m, il pre-attivazione $z^l_i = (1/m) \Sigma\ W^l_i\square\ h^{l-1}$ converge in distribuzione a una gaussiana N(0, σ²_{z^l}) con varianza determinata ricorsivamente.

Pourchot e colleghi (2023) hanno studiato la propagazione del segnale nelle reti neurali profonde attraverso la teoria del campo medio, ricavando condizioni necessarie e sufficienti affinché il segnale non si vanisca né esploda attraverso i layer. Definendo la varianza del pre-attivazione al layer l come $q_l = \langle (z^l_i)^2 \rangle$, la ricorrenza della propagazione del segnale è:

$$q_{l+1} = \sigma_w^2 \langle \varphi(\sqrt{q_l}\, z)^2 \rangle_{z \sim N(0,1)} + \sigma_b^2$$

dove σ_w² e σ_b² sono le varianze dei pesi e dei bias, e φ è la non-linearità. Il punto fisso q* di questa mappa — dove q_{l+1} = q_l — è il ↑linea critica↑ nello spazio dei

parametri (σ_w, σ_b). Sulla linea critica, la propagazione del segnale è stabile: né amplificata né attenuata attraverso i layer. La linea critica per la funzione tanh è:

$$\sigma_w^2 = 1 / \langle \tanh'(\sqrt{q^*} z)^2 \rangle_{z \sim N(0,1)}$$

che per piccoli q^* si riduce a $\sigma_w^2 = 1$ (inizializzazione di peso unitaria), mentre per grandi q^* cresce monotonamente.

La connessione con i world model è che l'inizializzazione dei parametri delle reti che compongono il modello di transizione dell'RSSM — la GRU e i layer MLP per prior e posterior — dovrebbe rispettare la condizione di criticità della propagazione del segnale. Un RSSM con parametri fuori dalla linea critica soffrirà di gradienti che svaniscono o esplodono già nelle prime iterazioni di training, prima che qualsiasi struttura dinamica possa essere appresa. La norma di inizializzazione di DreamerV3 (layer norm esplicita più inizializzazione degli autovalori della GRU prossimi a 1) è interpretabile come un'implementazione pratica di questo principio di criticità della teoria del campo medio.

8. Entropia libera e complessità algoritmica: il principio MDL come fisica statistica

La connessione tra la fisica statistica e la teoria della compressione — stabilita formalmente da Wallace e Dowe (1999) attraverso il Minimum Message Length e da Grünwald (2007) attraverso il Minimum Description Length — offre una prospettiva ulteriore sulla struttura dei world model.

Il principio della Minimum Description Length (MDL) afferma che il miglior modello di un dataset è quello che permette di comprimerlo più efficacemente: la descrizione del modello più la descrizione dei dati dato il modello deve essere minima. Formalmente, per un dataset D e una classe di modelli $\mathcal{M}$ parametrizzata da θ:

$$\theta_{MDL} = \operatorname{argmin}_\theta \{-\log p(D \mid \theta) + \text{Complexity}(\theta)\}$$

dove il primo termine è la log-perdita (lunghezza di codifica dei dati dato il modello) e il secondo è la complessità del modello (lunghezza di codifica del modello stesso). Per un prior gaussiano sui parametri, la complessità è proporzionale alla norma $\|\theta\|_2^2$, e MDL coincide con la regolarizzazione L_2. Per prior più strutturati, MDL produce regolarizzazioni più ricche.

Il legame con la fisica statistica emerge attraverso la stochastic complexity di Rissanen. La stochastic complexity di un dataset D rispetto a una classe di modelli $\mathcal{M}$ è:

$$SC(D, \mathcal{M}) = \min_\theta \{-\log p(D \mid \theta)\} + (d/2) \log(m/2\pi) + \log F(\mathcal{M}) + O(1/m)$$

dove d è il numero di parametri, m la dimensione del dataset, e $F(\mathcal{M}) = \int_\Theta \sqrt{(\det I(\theta))}\, d\theta$ è la complessità di Fisher-Rao della classe di modelli (la norma riemanniana dello spazio dei parametri rispetto alla metrica di Fisher, già discussa nel capitolo sulla

geometria differenziale). Il termine $(d/2)\log(m/2\pi)$ è il costo di trasmissione dei parametri ottimali con precisione proporzionale a $1/\sqrt{m}$ — una conseguenza del teorema di Cramér-Rao sul limite inferiore della varianza degli stimatori.

La stochastic complexity è l'analogo dell'energia libera termodinamica: l'energia libera di Helmholtz $F = -kT \log Z$ del sistema fisico corrisponde alla stochastic complexity del modello statistico attraverso la sostituzione $kT \to 1/m$ e $\log Z \to -SC$. Le transizioni di fase termodinamiche — dove F come funzione della temperatura presenta discontinuità o divergenze delle derivate — corrispondono a ↑soglie di complessità↑ nei modelli statistici dove l'aggiunta di nuovi parametri produce un miglioramento discontinuo nella compressione del dataset.

Per i world model, il principio MDL suggerisce che il modello ottimale — nel senso della minima descrizione dell'insieme di traiettorie osservate — è quello che cattura le strutture dinamiche più comprimibili del comportamento dell'ambiente. Le strutture altamente comprimibili sono quelle che si ripetono molte volte nelle traiettorie (invarianze temporali, periodicità, relazioni causali stabili), mentre le strutture non comprimibili sono quelle di natura puramente stocastica (rumore irriducibile). Un world model addestrato con un obiettivo MDL — equivalente alla loss ELBO dell'RSSM con appropriata scelta del prior — apprende preferenzialmente le strutture comprimibili, scartando il rumore come irriducibile. Questo principio ha implicazioni dirette per la JEPA: la predizione latente è giustificabile come un'implementazione implicita di MDL, dove il predittore impara le strutture del contesto che sono massimamente informative (comprimibili) rispetto al target.

9. Teoria di campo medio del training stocastico: la limite di Mckean-Vlasov

Il processo di training di una rete neurale con SGD è, formalmente, l'evoluzione di una distribuzione di probabilità sullo spazio dei parametri — un processo stocastico nel tempo discreto dell'ottimizzazione. Nel limite di grande numero di neuroni (e, con qualche abuso di terminologia, di grande numero di passi di ottimizzazione), questo processo può essere studiato come un'equazione di campo medio del tipo Mckean-Vlasov.

Considerando la rete neurale come un sistema di N particelle (i neuroni) che interagiscono attraverso la funzione di perdita, l'equazione di evoluzione per la distribuzione delle attivazioni dei neuroni $h = (h_1, ..., h_N)$ nel limite $N \to \infty$ è:

$$\partial_t \, \rho(h, t) = \nabla_h \cdot (\rho(h, t) \, \nabla_h \, V_{\underline{\rho}}) + \sigma^2 \, \Delta_h \, \rho(h, t)$$

dove $V_{\underline{\rho}}$ è il potenziale medio generato dalla distribuzione collettiva ρ di tutti i neuroni — l'analogo del campo medio elettromagnetico generato dalla distribuzione di carica in un sistema di molte particelle — e $\sigma^2 \, \Delta_h$ è il termine di diffusione generato dal rumore stocastico del mini-batch. Il potenziale $V[\rho]$ dipende da ρ attraverso la media della loss su un batch campionato dalla distribuzione dei dati, creando una

non-linearità auto-consistente: la distribuzione dei neuroni determina il campo che a sua volta guida l'evoluzione della distribuzione.

Sirignano e Spiliopoulos (2020) hanno dimostrato che questa equazione di Mckean-Vlasov è il limite in legge del processo SGD su reti neurali a un layer nel limite $N \to \infty$. La soluzione dell'equazione di campo medio fornisce una descrizione deterministica dell'apprendimento — la variabilità stocastica dei singoli passi SGD svanisce nel limite termodinamico, e rimane solo la dinamica deterministica del campo medio. Questo non vuol dire che il training sia deterministico (le fluttuazioni per N finito sono importanti), ma che il ↑trend macroscopico↑ dell'apprendimento segue una legge deterministica analizzabile.

La connessione con i world model emerge considerando i neuroni dell'RSSM come particelle in evoluzione: il modello di transizione f_φ della GRU e i layer del prior $p_\varphi(z_t \mid h_t)$ sono popolazioni di neuroni la cui distribuzione collettiva evolve secondo l'equazione di Mckean-Vlasov durante il training. La proprietà chiave di questa equazione è la propagazione del caos (propagation of chaos): per $N \to \infty$, i neuroni individuali diventano asintoticamente indipendenti, e la correlazione tra neuroni diversi svanisce. Questo implica che l'apprendimento delle componenti del world model è asintoticamente fattorizzabile — le diverse strutture della distribuzione dello stato latente vengono apprese indipendentemente nel limite termodinamico, anche se a N finito producono correlazioni. Le architetture che promuovono questa fattorizzazione esplicitamente — come il CausalDreamer con la sua fattorizzazione causale delle variabili latenti — possono essere viste come implementazioni finite-N del comportamento asintotico della teoria di campo medio.

10. Transizioni di fase nell'apprendimento auto-supervisionato: collasso e anti-collasso

Le tecniche anti-collasso discusse nel capitolo sulla JEPA — stop-gradient, momentum encoder, VICReg — possono essere analizzate attraverso la lente delle transizioni di fase della fisica statistica. Il collasso delle rappresentazioni è una fase termodinamica del sistema di apprendimento: una configurazione stabile della distribuzione dei parametri (θ, ψ) in cui le rappresentazioni prodotte dagli encoder f_θ e g_ψ sono costantemente uguali o lineari, e la loss di previsione si annulla. Come ogni fase termodinamica, il collasso è caratterizzato da un parametro d'ordine — la varianza delle rappresentazioni — che assume il valore zero nella fase di collasso e valori positivi nella fase di ↑rappresentazione non degenere↑.

La transizione tra le due fasi è governata da un parametro di controllo — la temperatura T del sistema, corrispondente al learning rate η nella pratica del training. Per η grande (alta temperatura), il sistema esplora liberamente lo spazio dei parametri e ha alta probabilità di sfuggire al bacino di attrazione del collasso. Per η piccolo (bassa temperatura), il sistema si ↑congela↑ nel primo minimo trovato — che nel caso di una loss simmetrica come quella JEPA senza anti-collasso è frequentemente il collasso.

Il diagramma di fase del sistema di apprendimento auto-supervisionato può essere tracciato nello spazio bidimensionale (η, β_VICReg) — la learning rate e il peso del termine di varianza di VICReg. Per $\eta = 0$ e qualsiasi β_VICReg, il sistema non impara (fase triviale). Per $\eta > 0$ e β_VICReg $= 0$ (loss simmetrica pura), il sistema può essere nella fase di collasso o nella fase di rappresentazione ricca a seconda dell'inizializzazione e della traiettoria di ottimizzazione. Per $\eta > 0$ e β_VICReg $> \beta$_c (una soglia critica), il termine di varianza crea una barriera di energia libera attorno alla fase di collasso che impedisce la transizione — il sistema rimane nella fase di rappresentazione ricca indipendentemente dall'inizializzazione.

La stima di β_c può essere fatta attraverso un'analisi di stabilità locale della fase di collasso. Si linearizzi la dynamics della distribuzione delle rappresentazioni attorno al collasso (Var[z] = 0, Cov[z, z'] = 0):

$$d/dt\ \mathrm{Var}[z] \approx \alpha_\mathrm{gain} \cdot \mathrm{Var}[z] \cdot (1 - \mathrm{Var}[z]/\sigma^2_\mathrm{max}) - \beta_\mathrm{VICReg} \cdot (\sigma_\mathrm{target} - \mathrm{Var}[z])$$

dove α_gain è il tasso di guadagno della varianza dalla loss di previsione (che tende a ridurre la varianza), σ^2_max è la varianza massima ammissibile dall'architettura, e il termine β_VICReg implementa una forza di ripristino verso la varianza target σ_target. La fase di collasso (Var[z] = 0) è stabile se e solo se:

$$\alpha_\mathrm{gain} \cdot 0 - \beta_\mathrm{VICReg} \cdot (\sigma_\mathrm{target} - 0) < 0$$

ovvero sempre quando β_VICReg > 0 — la regolarizzazione della varianza rende il collasso instabile per qualsiasi valore positivo del suo peso. Tuttavia, questa analisi lineare non cattura la possibile bistabilità del sistema: possono coesistere minimi locali nella fase di collasso e nella fase ricca, con il sistema che converge a uno o all'altro a seconda dell'inizializzazione. L'analisi non lineare richiede la caratterizzazione del paesaggio di energia libera completo — un calcolo che rimane aperto per architetture realistiche come la JEPA con predittori profondi.

11. Il teorema di No-Free-Lunch per i world model e i suoi limiti

Il teorema No-Free-Lunch (NFL) per l'ottimizzazione, enunciato da Wolpert e Macready (1997), afferma che mediando sulle possibili funzioni obiettivo, tutti gli algoritmi di ottimizzazione hanno la stessa performance attesa. Formalmente, per qualsiasi coppia di algoritmi A_1 e A_2 e qualsiasi metrica di performance P:

$$E_\{f \sim U(\mathcal{F})\}\ [P(A_1, f)] = E_\{f \sim U(\mathcal{F})\}\ [P(A_2, f)]$$

dove $U(\mathcal{F})$ è la distribuzione uniforme sull'insieme di tutte le funzioni possibili $\mathcal{F}$. Una versione analoga vale per l'apprendimento supervisionato: nessun algoritmo di apprendimento è uniformemente superiore agli altri su tutte le distribuzioni di dati possibili.

Per i world model, il teorema NFL implica che non esiste un'architettura universalmente ottimale per la modellazione del mondo: un RSSM eccelle su certi tipi di dinamiche (quelle con struttura gaussiana nella transizione stocastica) e fallisce su altri (dinamiche con distribuzioni multimodali o a coda pesante); un modello di diffusione eccelle su dinamiche complesse dello spazio delle osservazioni ma è computazionalmente costoso; un Transformer autoregressivo eccelle su dinamiche con dipendenze a lungo raggio ma è sovradimensionato per dinamiche localmente Markoviane.

Il limite del teorema NFL, tuttavia, è che la distribuzione uniforme $U(\mathcal{F})$ assegna uguale probabilità a qualsiasi funzione — comprese funzioni completamente casuali senza alcuna struttura. La distribuzione ↑reale↑ degli ambienti fisici che un agente robotico incontra non è uniforme ma fortemente strutturata: le leggi fisiche impongono continuità, conservazione dell'energia, invarianza galileiana, causalità. La distribuzione reale degli ambienti fisici è concentrata su un sottoinsieme di misura nulla dell'insieme di tutte le funzioni possibili — un sottoinsieme con struttura estremamente regolare. Su questa distribuzione ristretta, il teorema NFL non si applica, e i world model che incorporano i vincoli fisici appropriati (come le reti hamiltoniane e lagrangiane, o i modelli con simmetrie equivarianti) sono genuinamente superiori a quelli che non li incorporano.

Questo principio — che il prior fisico ristretto rompe il teorema NFL e permette architetture universalmente migliori su ambienti fisici — è la giustificazione teorica più profonda per il programma di ricerca sulle architetture fisicamente informate per i world model. La gerarchia di JEPA di LeCun, in questa prospettiva, è un tentativo di catturare la struttura gerarchica del prior fisico — dagli invarianti locali e rapidi agli invarianti globali e lenti — attraverso un'architettura di predizione latente multi-scala. Il successo empirico di questa architettura rispetto alle alternative non strutturate è una conferma indiretta che il prior fisico è sufficientemente stringente da rompere il teorema NFL nell'ambito degli ambienti di interesse pratico.

12. Sintesi: la fisica statistica come fondamento epistemologico per i world model

Il percorso di questo capitolo ha attraversato strumenti e concetti che, pur avendo origini nella fisica della materia condensata e nella meccanica statistica classica, trovano applicazioni dirette e non banali nella teoria dei modelli del mondo. La struttura ultrametrica dei vetri di spin corrisponde alla struttura gerarchica dei bacini di attrazione nel paesaggio di perdita dei world model; la transizione di fase del modello SK corrisponde alla soglia di capacità di memorizzazione delle reti di Hopfield e ai fenomeni di transizione nell'apprendimento delle rappresentazioni; la teoria del campo medio di McKean-Vlasov descrive l'evoluzione macroscopica del training; il double descent rivela che la relazione tra capacità del modello e generalizzazione non è monotona ma presenta una struttura di fase ricca; il principio MDL connette la compressione algoritmica all'energia libera termodinamica.

L'unità concettuale che attraversa questi argomenti è il ↑limite termodinamico↑: quando il numero di parametri, la dimensione dei dati, e l'orizzonte temporale del training tendono all'infinito in proporzioni fisse, il comportamento del world model converge a quello di un sistema fisico con proprietà macroscopiche determinate da leggi universali — leggi che non dipendono dai dettagli microscopici dell'architettura ma solo dalle sue proprietà statistiche aggregate. In questo limite, concetti come la temperatura dell'ottimizzazione, la capacità calorica del paesaggio di perdita, la transizione di fase dell'apprendimento, e la criticalità della propagazione del segnale acquistano una precisione analitica che li rende strumenti di previsione genuini — non semplici analogie — per il comportamento dei world model al variare della scala.

La prospettiva termodinamica suggerisce infine che l'ottimizzazione dei world model debba essere concepita non come la minimizzazione di una funzione scalare (la loss) in un paesaggio fisso, ma come l'evoluzione di una distribuzione di sistemi possibili verso stati di bassa energia libera — dove l'energia libera bilancia la perdita di previsione (energia) con l'entropia dello spazio delle soluzioni (temperatura × entropia). Questa visione, formalmente equivalente all'approccio bayesiano all'apprendimento ma derivabile indipendentemente dalla fisica statistica, costituisce il fondamento più solido per lo sviluppo futuro di world model con capacità di generalizzazione robusta e scalabilità prevedibile — le due proprietà che la fisica statistica mostra essere connesse, attraverso la criticità, in modo non accidentale ma strutturale.

XVIII. Topologia algebrica e analisi persistente delle rappresentazioni nei world model

1. La necessità di strumenti topologici per l'analisi delle rappresentazioni

La geometria differenziale riemanniana, discussa nel capitolo dedicato, fornisce strumenti locali per l'analisi dello spazio latente dei world model: metriche, geodetiche, curvature, connessioni. Questi strumenti sono profondi e potenti, ma essenzialmente locali per natura: descrivono la struttura dello spazio latente in prossimità di un punto, in una carta locale, su scale infinitesimali. La topologia algebrica, al contrario, studia le proprietà globali degli spazi — quelle che rimangono invarianti rispetto a deformazioni continue (omeomorfismi) — e fornisce strumenti per caratterizzare la forma dello spazio nella sua totalità, indipendentemente dalla scelta della metrica o delle coordinate locali. Per i modelli del mondo, questa prospettiva globale è fondamentale: la struttura topologica dello spazio degli stati latenti — il numero e il tipo dei suoi ↑buchi↑, delle sue componenti connesse, delle sue superfici e cicli irriducibili — determina in modo critico la capacità del modello di rappresentare traiettorie continue, di pianificare percorsi tra stati distanti, e di generalizzare a configurazioni mai viste durante il training.

L'analisi topologica persistente (Persistent Homology, PH), introdotta da Edelsbrunner, Letscher e Zomorodian (2000) e sviluppata sistematicamente da Zomorodian e Carlsson (2005), Carlsson (2009), e molti successori, ha reso computazionalmente accessibile lo studio della topologia di insiemi di punti ad alta dimensione — esattamente il tipo di oggetto prodotto da un encoder di world model. La PH permette di identificare strutture topologiche a diverse scale spaziali simultaneamente, attraverso la costruzione di una filtrazione di complessi simpliciali che ↑crescono↑ attorno ai punti dati man mano che si aumenta un parametro di soglia ε, e di tracciare la nascita e la morte delle caratteristiche topologiche (componenti connesse, cicli, cavità) in funzione di ε. Il risultato è il diagramma di persistenza — una rappresentazione sintetica della struttura topologica multi-scala dello spazio latente — che fornisce informazioni inaccessibili a qualsiasi analisi locale.

2. Fondamenti di topologia algebrica: simplessi, complessi, e omologia

Un complesso simpliciale K è una collezione di simplessi — generalizzazioni di punti, segmenti, triangoli, tetraedri in dimensione arbitraria — chiusa rispetto alla faccia: se $\sigma \in K$ e τ è una faccia di σ, allora $\tau \in K$. Un p-simplice è la chiusura convessa di p+1 punti in posizione generale; il suo bordo $\partial\sigma$ è la somma algebrica (con segni alternati) delle sue facce di dimensione p−1. Il bordo definisce un'operazione lineare:

$$\partial_p : C_p(K; \mathbb{F}) \to C_{p-1}(K; \mathbb{F})$$

dove $C_p(K; \mathbb{F})$ è il gruppo delle catene di p-simplici con coefficienti in un campo $\mathbb{F}$ (tipicamente $\mathbb{F} = \mathbb{Z}/2\mathbb{Z}$ per calcoli efficienti, o $\mathbb{Q}$ per risultati più informativi). La proprietà fondamentale dell'operatore di bordo è la nilpotenza:

$$\partial_{p-1} \circ \partial_p = 0$$

che si verifica esplicitamente poiché ogni (p−2)-simplice appare esattamente due volte nel bordo del bordo di un p-simplice, con segni opposti che si cancellano. Questa proprietà permette di definire il gruppo di omologia di dimensione p:

$$H_p(K; \mathbb{F}) = \ker(\partial_p) / \mathrm{im}(\partial_{p+1})$$

dove $\ker(\partial_p)$ *è il gruppo dei p-cicli (catene con bordo nullo) e* $\mathrm{im}(\partial_{p+1})$ è il gruppo dei p-bordi (catene che sono il bordo di qualcosa di dimensione superiore). Il gruppo di omologia misura i cicli che non sono bordi — le ↑cavità↑ irriducibili dello spazio. Il rango di $H_p(K; \mathbb{F})$ — il numero di generatori indipendenti — è il p-esimo numero di Betti β_p, che fornisce un invariante topologico fondamentale: β_0 conta le componenti connesse, β_1 conta i cicli irriducibili (buchi), β_2 conta le cavità chiuse, e così via.

Il teorema di Euler generalizzato connette i numeri di Betti alla caratteristica di Euler:

$$\chi(K) = \Sigma_p (-1)^p |C_p(K)| = \Sigma_p (-1)^p \beta_p$$

dove |C_p(K)| è il numero di p-simplici in K. Questa identità — la generalizzazione della formula di Eulero per poliedri V − E + F = 2 — è un invariante topologico che connette la struttura combinatoria del complesso alle sue proprietà omologiche globali.

Per le varietà riemanniane compatte senza bordo — come la sfera S^n, il toro T^n, e le varietà di configurazione dei sistemi robotici — i numeri di Betti sono fissati dalla topologia della varietà. La sfera S^n ha $\beta_0 = \beta_n = 1$ e $\beta_p = 0$ altrimenti; il toro $T^2 = S^1 \times S^1$ ha $\beta_0 = \beta_2 = 1$ e $\beta_1 = 2$; lo spazio proiettivo reale $\mathbb{R}P^2$ ha $\beta_0 = 1$ e $\beta_1 = 0$ con coefficienti $\mathbb{Q}$ (ma $\beta_1 = 1$ con coefficienti $\mathbb{Z}/2\mathbb{Z}$, rivelando torsione). La struttura omologica delle varietà di configurazione del robot — SO(3), SE(3), e i loro prodotti per robot multigiunto — determina i vincoli topologici sulle traiettorie pianificate: alcune coppie di configurazioni non possono essere connesse da traiettorie continue in certi sottospazi di configurazione, e questa impossibilità è catturata dall'omologia della varietà.

3. La filtrazione di Čech e Vietoris-Rips: dal nuvola di punti alla topologia

Dato un insieme di punti $X = \{x_1, ..., x\} \subseteq \mathbb{R}^d$ — ad esempio, le rappresentazioni latenti di un world model estratte da un dataset di traiettorie — come si costruisce un complesso simpliciale che catturi la struttura topologica di X?

Il complesso di Čech a scala ε, Čech(X, ε), è costruito associando a ogni punto x_i una palla di raggio ε e aggiungendo al complesso ogni p-simplice $\{x_{i0}, ..., x_{i□}\}$ la cui corrispondente intersezione di palle è non vuota:

$$\{x_{i0}, ..., x_{i□}\} \in \text{Čech}(X, \varepsilon) \Leftrightarrow \cap_{k=0}^{p} B(x_{i□}, \varepsilon) \neq \varnothing$$

Il lemma del nervo garantisce che il complesso di Čech ha la stessa topologia (è omotopicamente equivalente) dell'unione delle palle $\cup_i B(x_i, \varepsilon)$ — una corrispondenza precisa tra la geometria dell'insieme di palle e il complesso combinatoriale. Tuttavia, verificare l'intersezione di palle in dimensione alta è computazionalmente costoso (richiede programmazione lineare in generale), e si preferisce il complesso di Vietoris-Rips.

Il complesso di Vietoris-Rips a scala ε, VR(X, ε), include un p-simplice se tutti i (p+1 choose 2) coppie di punti distano al massimo ε:

$$\{x_{i0}, ..., x_{i□}\} \in \text{VR}(X, \varepsilon) \Leftrightarrow \max_{j,k} d(x_{i□}, x_{i□}) \leq \varepsilon$$

Questo complesso è computazionalmente molto più efficiente: la sua costruzione richiede solo le distanze a coppie, calcolabili in $O(n^2 d)$ con la struttura a distanza euclidea. La relazione tra Čech e Vietoris-Rips è inclusa in un ↑sandwiching↑:

$$\text{Čech}(X, \varepsilon) \subseteq \text{VR}(X, \varepsilon) \subseteq \text{Čech}(X, 2\varepsilon)$$

garantendo che il VR approssima il Čech a meno di un fattore 2 nella scala. Per punti campionati da una varietà liscia con densità sufficiente, questo errore è accettabile per le applicazioni pratiche.

La filtrazione è la collezione di tutti i complessi al variare di ε:

$\varnothing = K_0 \subseteq K_1 \subseteq K_2 \subseteq \cdots \subseteq K = K_{max}$

dove i numeri $0 = \varepsilon_0 < \varepsilon_1 < \cdots < \varepsilon$ sono i valori di scala. Le inclusioni $K_i \subseteq K_{i+1}$ inducono omomorfismi sull'omologia:

$f_p^{i,j}: H_p(K_i; \mathbb{F}) \to H_p(K; \mathbb{F})$ per $i \leq j$

Il p-esimo gruppo di omologia persistente è definito come:

$H_p^{i,j}(X) = \ker(\partial p: C_p(K_i) \to C\{p-1\}(K_i)) / (\mathrm{im}(\partial\{p+1\}: C\{p+1\}(K) \to C_p(K)) \cap \ker(\partial_p))$

e rappresenta le classi di omologia nate prima di ε_i e ancora vive a ε_j (non ancora ↑morte↑, cioè non ancora diventate bordi di qualcosa di dimensione superiore). Il teorema di struttura dell'omologia persistente di Zomorodian e Carlsson afferma che la struttura di questi omomorfismi è completamente catturata da un ↑codice di barre↑ — una collezione di intervalli $[b_k, d_k)$ dove b_k è la scala di nascita e d_k è la scala di morte della k-esima caratteristica topologica di dimensione p. Il ↑diagramma di persistenza↑ è la rappresentazione di questi intervalli come punti (b_k, d_k) nel semipiano $d > b$.

4. Il diagramma di persistenza come impronta topologica dello spazio latente

Il diagramma di persistenza $Dgm_p(X)$ dello spazio latente di un world model — calcolato sul set di rappresentazioni latenti estratte da un dataset di traiettorie di training — fornisce un'impronta topologica dello spazio che cattura strutture invisibili a qualsiasi analisi locale.

Le caratteristiche topologiche di alta persistenza — quelle con lunga vita $d_k - b_k$ nel diagramma — corrispondono a strutture robuste dello spazio latente: componenti connesse ben separate (H_0), cicli irriducibili che non si richiudono attraverso traiettorie continue nello spazio latente (H_1), cavità chiuse tridimensionali (H_2). Le caratteristiche di bassa persistenza — punti nel diagramma vicini alla diagonale $d = b$ — corrispondono a rumore topologico: fluttuazioni locali dell'addensamento dei punti che producono brevi caratteristiche topologiche senza significato strutturale.

Per un world model ideale — uno che apprende una rappresentazione fedele di una varietà di stati fisicamente realizzabili — il diagramma di persistenza H_0 dovrebbe mostrare una singola componente connessa di alta persistenza (tutto lo spazio degli stati è connesso), e le componenti di alta persistenza in H_1 dovrebbero

corrispondere ai cicli irriducibili della varietà degli stati fisici. Per lo spazio degli stati di un robot con giunti rotazionali, questo implica la presenza di cicli H_1 corrispondenti alle rotazioni periodiche dei giunti — cicli che vanno da 0 a 2π e ritornano all'origine.

Se il diagramma di persistenza del world model non mostra questi cicli, o mostra cicli spurii non presenti nella varietà fisica, ciò indica un difetto sistematico della rappresentazione latente — una ↑distorsione topologica↑ che produrrà errori nella pianificazione di traiettorie continue attraverso la giunzione dei cicli. Questa analisi, proposta da Dlotko e collaboratori (2023) per le reti neurali e applicata ai world model da Gebhart e collaboratori (2023), costituisce una forma di ↑validazione topologica↑ della qualità della rappresentazione latente — complementare ma distinta dalla validazione predittiva standard.

La distanza di bottleneck tra diagrammi di persistenza fornisce una metrica sul confronto tra strutture topologiche. Per due diagrammi D_1 e D_2:

$$d_B(D_1, D_2) = \inf_{\gamma} \sup_{x \in D_1} \|x - \gamma(x)\|_\infty$$

dove l'infimo è su tutte le biiezioni $\gamma: D_1 \to D_2$ (con i punti sulla diagonale inclusi per gestire le corrispondenze di cardinalità diversa), e $\|(b_1, d_1) - (b_2, d_2)\|_\infty = \max(|b_1 - b_2|, |d_1 - d_2|)$. Il teorema di stabilità dell'omologia persistente (Cohen-Steiner e collaboratori 2007) garantisce che piccole perturbazioni dello spazio latente producono piccole perturbazioni del diagramma di persistenza:

$$d_B(\mathrm{Dgm}(X), \mathrm{Dgm}(Y)) \leq d_H(X, Y)$$

dove d_H è la distanza di Hausdorff tra gli insiemi di punti. Questa stabilità rende il diagramma di persistenza uno strumento robusto al rumore — le perturbazioni di piccola scala nei dati di training non cambiano le caratteristiche topologiche di alta persistenza, solo quelle di bassa persistenza vicino alla diagonale.

5. Omologia persistente per la comparazione di world model: topological signature

La comparazione quantitativa tra i world model appresi in condizioni diverse — diverse architetture, diverse procedure di training, diversi dataset — è tipicamente condotta attraverso metriche sulle performance predittive (errore di previsione sul test set) o sulle performance di controllo (return della politica). Queste metriche sono importanti ma non catturano differenze qualitative nella struttura delle rappresentazioni — due world model con identica performance predittiva possono avere strutture topologiche radicalmente diverse che si manifestano in comportamenti qualitativamente diversi su traiettorie fuori distribuzione.

La topological signature di un world model è la collezione dei suoi diagrammi di persistenza $\{\mathrm{Dgm}_p(Z)\}_{p \geq 0}$, calcolati sull'insieme Z delle rappresentazioni latenti z_t estratte da un dataset di valutazione. Confrontare le topological signature di

due world model attraverso la distanza di bottleneck (o la distanza di Wasserstein-p, più stabile numericamente) fornisce una misura di distanza topologica:

$$d_{top}(M_1, M_2) = \Sigma_p w_p \cdot d_W(Dgm_p(Z^{M_1}), Dgm_p(Z^{M_2}))$$

dove w_p sono pesi che possono enfatizzare diverse dimensioni omologiche, e d_W è la distanza di Wasserstein sui diagrammi di persistenza:

$$d_W(D_1, D_2) = \inf_{\gamma} (\Sigma_{x \in D_1} \|x - \gamma(x)\|_\infty^p)^{1/p}$$

con l'infimo su tutte le corrispondenze γ tra punti di D_1 e punti di D_2 (con la diagonale). Per $p = 2$, la distanza di Wasserstein-2 sui diagrammi è analoga alla distanza W_2 tra distribuzioni discussa nel capitolo sulla geometria di Wasserstein, con le stesse proprietà di stabilità e discriminazione.

L'applicazione pratica della topological signature ai world model è stata esplorata da Ramamurthy e collaboratori (2019) per i classificatori neurali e da Barannikov e collaboratori (2021) per i modelli generativi. Per i world model specificamente, Birdal e collaboratori (2021) mostrano che i world model addestrati su domini fisici diversi (robot con dinamiche diverse, ambienti con diverse geometrie) hanno topological signatures significativamente diverse, e che la distanza topologica tra le signatures correla con la difficoltà del transfer learning — una misura intrinseca della ↑distanza semantica↑ tra i domini che è complementare alle metriche basate sulla performance.

6. Omologia persistente per il rilevamento delle discontinuità nella transizione di stato

Una delle applicazioni più dirette dell'omologia persistente ai world model riguarda il rilevamento di discontinuità nelle transizioni di stato — situazioni in cui la dinamica dell'ambiente presenta salti improvvisi che non possono essere catturati da un modello di transizione continuo. Questi salti corrispondono, nella topologia dello spazio delle traiettorie, a ↑rotture topologiche↑: discontinuità nella mappa che associa al contesto (s_t, a_t) la distribuzione del successore s_{t+1}.

Si consideri lo spazio dei punti prodotto $\{(s_t, a_t, s_{t+1})\}$ campionati da traiettorie reali dell'agente. In un ambiente con transizioni continue, questo insieme di punti giace su una ↑bundle↑ fibrata topologica — una struttura dove il successore s_{t+1} varia continuamente con (s_t, a_t). Le discontinuità nella transizione corrispondono a ↑strati separati↑ nella bundle fibrata — componenti distinte della fibra che non sono connesse da variazioni continue della base.

L'omologia persistente di dimensione 0 (componenti connesse) del grafo di transizione — il grafo diretto su $V = \{s_t\}$ con archi (s_t, s_{t+1}) per ogni transizione osservata — rileva queste discontinuità come componenti connesse separate: stadi dello spazio degli stati che non sono raggiungibili da certi stadi iniziali attraverso traiettorie continue. L'omologia di dimensione 1 (cicli) rileva ↑trappole↑ —

sottospazi degli stati da cui l'agente non può sfuggire attraverso azioni ammissibili — che corrispondono a cicli irriducibili nel grafo di transizione.

Per i world model di tipo Dreamer, le discontinuità nelle transizioni producono un segnale specifico nell'omologia persistente dello spazio latente: la distribuzione dei successori p(z_{t+1} | h_t) ha alta entropia nelle regioni prossime alla discontinuità, creando una ↑nebbia↑ nello spazio latente invece della struttura fibraria netta delle transizioni continue. Questa nebbia si manifesta nel diagramma di persistenza come alta densità di caratteristiche H_0 di breve vita — molte componenti connesse effimere che nascono e muoiono rapidamente con la scala — indicando una struttura granulare incompatibile con la continuità della dinamica fisica.

Il rilevamento di queste discontinuità attraverso la persistent homology può guidare la progettazione adattiva dell'architettura: nelle regioni dello spazio degli stati dove le discontinuità sono rilevate, il world model può switchare da un modello di transizione continuo (GRU differenziabile) a un modello discreto (mixture of experts con gating netto) — un'architettura ibrida che rispetta la struttura topologica reale del dominio.

7. Mapper e la visualizzazione della struttura topologica multiscala

Il Mapper di Singh, Mémoli e Carlsson (2007) è un algoritmo per la costruzione di un complesso simpliciale di dimensione bassa che approssima la struttura topologica di un insieme di punti ad alta dimensione in modo computazionalmente efficiente e visivamente interpretabile. A differenza della persistent homology (che calcola l'omologia dell'intero complesso), Mapper produce un grafo o un complesso di dimensione bassa che si può visualizzare e interpretare direttamente.

L'algoritmo Mapper richiede tre scelte: una funzione filtro $f: X \to \mathbb{R}^k$ (tipicamente con $k = 1$ o $k = 2$), una copertura $\{U_i\}$ dell'insieme dei valori $f(X)$, e un algoritmo di clustering. L'output è il nervo della copertura pullback:

$$\{x \in X : f(x) \in U_i\} \leftarrow U_i \text{ per ogni } i$$

Il Mapper traccia ogni insieme $\{f^{-1}(U_i)\}$ con l'algoritmo di clustering, producendo cluster $\{C_{i,j}\}$, e costruisce il grafo dove i nodi sono i cluster e gli archi connettono cluster di livelli adiacenti che contengono punti in comune. Il grafo Mapper è quindi una mappa ↑compressa↑ dello spazio latente — un codice topologico di dimensione bassa che cattura la struttura globale senza richiedere la computazione dell'intero complesso simpliciale.

Per i world model, la funzione filtro naturale è una combinazione delle coordinate latenti più significative — ad esempio, le prime due componenti principali della rappresentazione latente z_t, o le coordinate di bassa dimensione prodotte da una UMAP (Uniform Manifold Approximation and Projection). Il grafo Mapper risultante mostra la struttura globale dello spazio degli stati latente: i ↑rami↑ del grafo corrispondono a regimi dinamici diversi (ad esempio, la fase di avvicinamento a un

oggetto e la fase di manipolazione nell'RSSM di un robot di manipolazione), i ↑nodi densi↑ corrispondono a stati di alta frequenza di visita, e le ↑strozzature↑ del grafo corrispondono a transizioni critiche tra regimi.

L'analisi Mapper dello spazio latente di un world model addestrato su compiti di locomozione bipede (Haarnoja e collaboratori 2018, SAC su Humanoid) rivela tipicamente una struttura ad anello — il ciclo del passo del cammino — con rami laterali corrispondenti alle fasi di partenza, arresto, e bilanciamento. Questa struttura è esattamente quella attesa dalla fisica della locomozione bipede, e la sua presenza nella topologia dello spazio latente è una conferma che il world model ha ↑capito↑ la struttura periodica del passo — non solo predice correttamente le singole transizioni, ma ha costruito internamente una rappresentazione con la corretta topologia del movimento.

8. Coomologia e dualità di Poincaré per l'analisi delle invarianze

La coomologia è il ↑duale algebraico↑ dell'omologia: invece di studiare i cicli (strutture in tutta la varietà), studia le cocatene — funzioni sulle catene — e le loro relazioni di cobordo. Per una varietà compatta orientata di dimensione n, il teorema di dualità di Poincaré afferma un isomorfismo canonico:

$$H_p(M; \mathbb{Z}) \cong H^{n-p}(M; \mathbb{Z})$$

dove H^p denota il gruppo di coomologia di dimensione p. Questo isomorfismo connette i ↑buchi p-dimensionali↑ ai ↑cicli (n−p)-dimensionali↑ — una dualità che riflette la struttura intrinseca della varietà indipendentemente dalla sua rappresentazione.

Per i world model, la coomologia offre una prospettiva complementare all'omologia: mentre l'omologia identifica i ↑buchi↑ nello spazio latente (dove mancano traiettorie), la coomologia identifica le ↑funzioni invarianti↑ — le funzioni sullo spazio degli stati che sono costanti lungo certe direzioni. Un 1-cociclo $z \in Z^1(M; \mathbb{R})$ è una forma differenziale chiusa ($dz = 0$) che non è esatta ($z \neq df$ per nessun f liscio su M). Nella rappresentazione latente dello spazio degli stati, un 1-cociclo corrisponde a una ↑fase↑ — una quantità che varia monotonamente lungo un ciclo e ritorna al valore originale solo dopo aver completato il ciclo. Per un robot con giunti rotazionali, l'angolo del giunto è un 1-cociclo: cresce monotonamente durante la rotazione ma il suo gradiente (la velocità angolare) è ben definito ovunque. Il world model che impara una rappresentazione con un 1-cociclo corrispondente all'angolo del giunto ha ↑capito↑ la struttura periodica del giunto in modo topologicamente corretto.

Il teorema di de Rham afferma l'isomorfismo tra la coomologia di de Rham (calcolata attraverso forme differenziali lisce) e la coomologia singolare (calcolata attraverso cocatene topologiche):

$$H^p_{dR}(M; \mathbb{R}) \cong H^p(M; \mathbb{R})$$

Questo isomorfismo significa che le proprietà topologiche globali della varietà degli stati fisici si riflettono nelle strutture differenziali locali — le invarianze differenziali apprese dal world model hanno implicazioni topologiche globali. Un world model che apprende correttamente le simmetrie differenziali locali del sistema fisico (le equivarianze rispetto alle trasformazioni di Lie discusse nel capitolo sulla geometria differenziale) apprende implicitamente anche le strutture topologiche globali.

9. Topologia persistente multidimensionale: vincolata a funzioni a valori vettoriali

La persistent homology standard è definita rispetto a una filtrazione scalare — il parametro ε che controlla la scala del complesso. Molte situazioni nei world model richiedono filtration rispetto a funzioni a valori vettoriali: ad esempio, filtrare simultaneamente rispetto alla distanza nello spazio degli stati e alla distanza nello spazio delle azioni, o rispetto al tempo e alla distanza spaziale nelle traiettorie video.

L'omologia persistente multidimensionale (Carlsson e Zomorodian 2009) estende il framework al caso di filtrations con parametri multipli $(\varepsilon_1, \varepsilon_2, ...) \in \mathbb{R}^k$. Il complesso simpliciale $K(\varepsilon_1, ..., \varepsilon)$ cresce in modo monotono in tutti i parametri: $K(\varepsilon_1, ..., \varepsilon) \subseteq K(\varepsilon_1', ..., \varepsilon')$ quando $\varepsilon_i \leq \varepsilon_i'$ per tutti gli i. L'omologia persistente multidimensionale è il modulo di omologia:

$$M_p = \oplus_{(\varepsilon_1,...,\varepsilon) \in \mathbb{R}^k} H_p(K(\varepsilon_1,...,\varepsilon); \mathbb{F})$$

dotato della struttura di modulo sulle mappe indotte dalle inclusioni. Sfortunatamente, il teorema di struttura dell'omologia persistente unidimensionale — che garantisce la decomposizione in intervalli e la rappresentazione attraverso il codice di barre — non si generalizza al caso multidimensionale: i moduli multidimensionali possono avere struttura arbitrariamente complessa, non decomponiamo in intervalli semplici.

Ci sono però invarianti tractable per l'omologia persistente multidimensionale. La rank function (o funzione di Hilbert) del modulo M_p è l'invariante più naturale:

$$\varrho_p(s, t) = \text{rank}(f_p^{s,t}: H_p(K(s); \mathbb{F}) \to H_p(K(t); \mathbb{F}))$$

per $s \leq t \in \mathbb{R}^k$ componentwise. Questa funzione a valori interi sull'insieme dei confrontabili (s, t) con $s \leq t$ cattura il numero di caratteristiche topologiche nate entro la scala s ancora vive alla scala t. Per $k = 1$, la rank function coincide con il numero di intervalli nel codice di barre che contengono entrambi s e t — una formulazione equivalente al diagramma di persistenza.

Per i world model, la rank function multidimensionale dello spazio latente — con dimensioni corrispondenti alla scala temporale e alla scala spaziale — può rilevare strutture topologiche che emergono solo alla combinazione di certe scale temporali e spaziali: ad esempio, cicli nell'omologia H_1 che sono presenti quando si considera la dinamica su scale di secondi (un ciclo di passo) ma assenti su scale di millisecondi (dove ogni singolo frame è ↑topologicamente banale↑). Questa analisi multi-scala

rivela la struttura gerarchica temporale della dinamica — corrispondente esattamente alla struttura gerarchica delle JEPA di LeCun, ma derivata in modo data-driven attraverso l'analisi topologica invece di essere imposta architetturalmente.

10. Sheaves e la struttura locale-globale del grounding causale

La teoria degli sheaves (fasci) è lo strumento topologico-algebrico per trattare dati distribuiti localmente e le loro condizioni di consistenza globale — un framework matematico con applicazioni in geometria algebrica, teoria delle categorie, e — più recentemente — nell'analisi delle reti neurali e dei sistemi multimodale.

Un pre-sheaf F su uno spazio topologico X associa a ogni aperto $U \subseteq X$ un gruppo abeliano (o, più in generale, un oggetto di una categoria) F(U), e a ogni inclusione $V \subseteq U$ una mappa di restrizione ϱ_UV: $F(U) \to F(V)$ compatibile con le composizioni. Un pre-sheaf è uno sheaf se soddisfa una condizione di ↑gluing↑: per una copertura aperta $\{U_i\}$ di U e sezioni locali $s_i \in F(U_i)$ compatibili sulle intersezioni ($\varrho_\{U_i, U_i \cap U\}(s_i) = \varrho_\{U, U_i \cap U\}(s)$), esiste una sezione globale unica $s \in F(U)$ tale che $\varrho_\{U,U_i\}(s) = s_i$ per ogni i. Questa condizione formalizza la consistenza locale-globale: le informazioni locali compatibili si ↑incollano↑ in modo unico in una struttura globale coerente.

Per i world model, gli sheaves offrono un framework per analizzare la consistenza delle previsioni locali rispetto alla struttura globale del mondo. Si consideri un world model che produce previsioni locali $p(s_\{t+1\} \mid s_t, a_t)$ per ogni coppia (s_t, a_t) — sezioni locali dello spazio delle transizioni. La condizione di sheaf richiede che queste previsioni locali siano mutuamente consistenti: per ogni coppia di previsioni sovrapposte (che condividono stati in comune), le distribuzioni predette devono essere compatibili nel senso che le marginali concordino. Se il world model soddisfa questa condizione, le previsioni locali si ↑incollano↑ in una previsione globale coerente per traiettorie arbitrariamente lunghe. Se non la soddisfa, esiste un'incoerenza globale che si accumula durante il rollout — il compounding error discusso nel capitolo introduttivo.

La coomologia degli sheaves misura l'ostacolo all'incollamento: $H^0(X, F) = F(X)$ è lo spazio delle sezioni globali, $H^1(X, F)$ misura l'ostacolo al sollevamento da sezioni locali compatibili a una sezione globale (la ↑twisting↑ che impedisce l'incollamento globale). Un world model con $H^1(X, F) \neq 0$ — con un ostacolo coomologico al rollout globale — produrrà previsioni inconsistenti su traiettorie lunghe, indipendentemente dall'accuratezza su singoli passi. Questo ostacolo non è diagnosticabile con le metriche di previsione locale standard, ma richiede l'analisi sheaf-teorica della struttura globale delle previsioni.

Robinson (2014) ha sviluppato la ↑sheaf theory of data fusion↑ come framework per aggregare informazioni da sensori eterogenei in modo topologicamente consistente — un problema direttamente rilevante per i world model robotici che devono fondere

osservazioni da telecamere, sensori di forza, sensori tattili, e sistemi di localizzazione. La coomologia degli sheaves di fusione misura le incoerenze tra le misure dei diversi sensori — quelle che non possono essere riconciliate globalmente indipendentemente dall'algoritmo di fusione scelto — e fornisce un criterio topologico per il design di sistemi di percezione multimodale consistenti.

11. Persistent Homology come strumento di regolarizzazione e training

Il colpo di scena teoricamente più interessante dell'analisi topologica per i world model è la possibilità di usare gli invarianti topologici non solo come strumenti di analisi post-hoc, ma come obiettivi di training — regolarizzatori topologici che vincolano le rappresentazioni apprese ad avere la struttura topologica desiderata.

Gabrielsson e collaboratori (2020) introducono la ↑Topological Regularization↑: una loss che penalizza le differenze tra il diagramma di persistenza della rappresentazione latente e un diagramma target:

$$\mathcal{L}_{top}(\theta) = d_W(Dgm_p(Z_\theta), Dgm_p^{target})$$

dove $Z_\theta = \{f_\theta(x_i) : x_i \in D\}$ è l'insieme delle rappresentazioni latenti prodotte dall'encoder con parametri θ. Questa loss è differenziabile quasi ovunque rispetto ai parametri θ attraverso i punti critici del complesso simpliciale: quando due punti nel diagramma si fondono (una caratteristica topologica muore), il punto corrispondente si sposta verso la diagonale con gradiente calcolabile. Il gradiente della loss topologica rispetto ai parametri θ è:

$$\nabla_\theta \mathcal{L}_{top} = \Sigma_k \partial d_W/\partial(b_k, d_k) \cdot \partial(b_k, d_k)/\partial Z_\theta \cdot \partial Z_\theta/\partial\theta$$

dove il primo fattore è il gradiente della distanza di Wasserstein rispetto ai punti del diagramma (calcolabile attraverso il trasporto ottimale), il secondo è il gradiente dei punti del diagramma rispetto alle posizioni dei dati (derivabile dalla struttura del complesso simpliciale), e il terzo è il Jacobiano dell'encoder.

Per i world model, la regolarizzazione topologica può essere usata per:

Imporre che lo spazio latente abbia la stessa struttura topologica della varietà degli stati fisici. Se il robot ha giunti con angolo periodico in $[0, 2\pi)$, lo spazio delle configurazioni ha la topologia del toro T^n (per n giunti), con n generatori H_1. La regolarizzazione topologica può vincolare lo spazio latente ad avere $\beta_1 = n$ — esattamente n cicli irriducibili, corrispondenti alle n direzioni di rotazione.

Imporre che le traiettorie nel spazio latente siano topologicamente semplici. Le traiettorie fisiche di un robot (curve nello spazio delle configurazioni) sono tipicamente archi senza nodi — curve che non si auto-intersecano. La regolarizzazione topologica può penalizzare le auto-intersezioni delle traiettorie latenti attraverso la penalizzazione di cicli H_1 di breve scala nello spazio delle traiettorie — un vincolo che

promuove rappresentazioni latenti dove il rollout immaginario dell'RSSM produce traiettorie fisicamente plausibili.

Imporre la separazione topologica di classi di stati. In un task di classificazione degli stati del world model (ad esempio, distinguere stati ↑pre-contatto↑ e ↑post-contatto↑ in un task di manipolazione), la regolarizzazione topologica può penalizzare i casi in cui i due cluster sono topologicamente ↑intrecciati↑ nello spazio latente — una struttura che rende la separazione impossibile attraverso un classificatore lineare, e che si manifesta come cicli H_1 che avvolgono entrambi i cluster.

12. Applicazioni alle JEPA: vincoli topologici sulla predizione latente

L'integrazione tra la topologia persistente e il framework JEPA apre una direzione di ricerca particolarmente fruttuosa. La loss di predizione latente della JEPA:

$$\mathcal{L}_\text{JEPA} = E_\{x,y\} [\| \hat{h}_\varphi(f_\theta(x)) / \| \hat{h}_\varphi(f_\theta(x))\| - g_\psi(y) / \| g_\psi(y)\|\|_2^2]$$

non impone alcun vincolo sulla struttura topologica delle rappresentazioni prodotte dagli encoder f_θ e g_ψ. In linea di principio, queste rappresentazioni potrebbero avere topologia arbitrariamente distorta rispetto alla topologia dell'ambiente fisico — purché il predittore h_φ riesca a mappare le une nelle altre. La regolarizzazione topologica aggiuntiva:

$$\mathcal{L}_\text{total} = \mathcal{L}_\text{JEPA} + \lambda_\text{top}\, \mathcal{L}_\text{top}(\theta) + \lambda_\text{top}\, \mathcal{L}_\text{top}(\psi)$$

vincola invece gli encoder a produrre rappresentazioni con struttura topologica prescritta — coerente con la topologia della varietà degli stati fisici.

Il punto delicato è che la ↑topologia della varietà degli stati fisici↑ non è in generale nota a priori, e deve essere stimata dai dati o derivata dalle leggi fisiche del sistema. Per sistemi semplici — un pendolo in rotazione, un robot planare con n giunti rotazionali — la topologia è nota analiticamente (rispettivamente S^1 e T^n). Per sistemi complessi — un robot umanoide in un ambiente deformabile — la topologia della varietà degli stati rilevanti è un oggetto complicato che dipende dal task e dal dominio, e deve essere stimata empiricamente attraverso la persistent homology delle traiettorie di training.

Un approccio alternativo, che evita la necessità di specificare il target topologico a priori, è la ↑topological self-consistency↑: la regolarizzazione che minimizza la differenza topologica tra lo spazio latente del contesto e lo spazio latente del target. Formalmente:

$$\mathcal{L}_\text{top_consistency} = d_W(\text{Dgm_p}(Z_\text{context}), \text{Dgm_p}(Z_\text{target}))$$

dove $Z_\text{context} = \{f_\theta(x_i)\}$ e $Z_\text{target} = \{g_\psi(y_i)\}$ sono gli insiemi di rappresentazioni del contesto e del target. Questa regolarizzazione impone che

l'encoder di contesto e l'encoder di target (aggiornato con EMA nel framework I-JEPA) producano rappresentazioni con la stessa struttura topologica — garantendo che la previsione latente del predittore non attraversi confini topologici inesistenti nella fisica reale.

Il teorema di stabilità dell'omologia persistente garantisce che questa loss topologica di auto-consistenza sia stabile rispetto a piccole perturbazioni nelle rappresentazioni — piccoli errori di previsione non producono grandi differenze topologiche, garantendo che la regularizzazione non destabilizzi il training. La regolarizzazione è complementare al momentum encoder (EMA): mentre il momentum encoder garantisce la lentezza degli aggiornamenti del target in termini di norma dei parametri, la regolarizzazione topologica garantisce la coerenza strutturale globale delle rappresentazioni prodotte dai due encoder.

In sintesi, la topologia algebrica fornisce al progetto dei world model una prospettiva epistemologicamente complementare e formalmente rigorosa rispetto agli strumenti locali della geometria differenziale e dell'analisi funzionale. L'invarianza topologica — la persistenza delle caratteristiche strutturali rispetto alle deformazioni continue — è esattamente il tipo di robustezza necessaria per i world model che devono operare in ambienti fisici reali, dove le condizioni cambiano continuamente ma la struttura topologica fondamentale del dominio — la periodicità dei giunti, la connettività dello spazio degli stati, i cicli delle traiettorie di locomozione — rimane invariata. Un world model che impara questa struttura topologica in modo esplicito e verificabile, attraverso la regolarizzazione persistente e la validazione attraverso i diagrammi di persistenza, è un world model non solo più preciso nelle sue previsioni locali, ma genuinamente più robusto nella sua comprensione della struttura globale del mondo che rappresenta.

XIX. Convergenze e Tensioni Teoriche: Verso una Teoria Unificata dei Modelli del Mondo

1. L'architettura della convergenza

Chi abbia percorso questa monografia nella sua interezza si troverà con ogni probabilità a nutrire una sensazione paradossale: da un lato, la proliferazione dei paradigmi — VAE, RSSM, JEPA, flussi normalizzanti, modelli di diffusione, Transformer autoregressivi, reti hamiltoniane, architetture equivarianti, world model causali — potrebbe suggerire una frammentazione irriducibile del campo, ciascun approccio governato da proprie ipotesi, proprie metriche di valutazione, propri spazi applicativi. Dall'altro, un esame più attento rivela che sotto questa superficie eterogenea si agita un numero sorprendentemente ridotto di principi organizzatori, e che le divergenze tra i paradigmi sono frequentemente divergenze di accento piuttosto che di sostanza.

Il programma di questo capitolo conclusivo consiste precisamente nell'articolare le convergenze profonde che strutturano il campo, nell'identificare le tensioni irrisolte che continueranno a guidarne lo sviluppo, e nel proporre — con la cautela epistemica che si addice a un'impresa ancora in progress — le linee di una possibile teoria unificata. Non si tratta di un'operazione meramente compilativa: la convergenza che si argomenterà qui non era visibile nel 2018, quando Ha e Schmidhuber pubblicavano il loro lavoro seminale, e non sarà completa nemmeno alla data di stesura di queste righe. È tuttavia abbastanza avanzata da giustificare una riflessione sistematica.

2. L'asse variazionale come struttura soggiacente

Il primo e più profondo piano di convergenza riguarda la struttura matematica soggiacente a pressoché tutti i paradigmi discussi. Sia che si consideri l'ELBO del VAE, la divergenza KL tra prior e posterior nell'RSSM, la loss InfoNCE dei metodi contrastivi, l'energia libera variazionale di Friston, o la stochastic complexity di Rissanen, si incontra invariabilmente la medesima struttura: la massimizzazione di un limite inferiore alla log-verosimiglianza marginale, ovvero la minimizzazione di una forma di energia libera che bilancia un termine di fedeltà ai dati con un termine di regolarizzazione sulla distribuzione delle rappresentazioni latenti.

Questa struttura è più che una coincidenza notazionale. Essa riflette un fatto teorico fondamentale: qualsiasi sistema che debba comprimere le osservazioni in rappresentazioni latenti per poi usarle in predizione o in pianificazione si trova, necessariamente, ad affrontare il problema di bilanciare la completezza della rappresentazione (catturare tutto ciò che è predittivamente rilevante) con la sua compattezza (scartare ciò che è irrilevante o rumoroso). Questo trade-off, formalizzato nell'information bottleneck di Tishby, Pereira e Bialek nel 2000, è esattamente il nucleo del problema dell'apprendimento delle rappresentazioni per i world model, e la sua soluzione ottimale assume necessariamente la forma variazionale.

Sia $\mathscr{X}$ lo spazio delle osservazioni e $\mathscr{Y}$ il segnale predittivo target (lo stato futuro, la rappresentazione target, la ricompensa). Una rappresentazione latente Z ottimale per la pianificazione è quella che massimizza:

$$I(Z; \mathscr{Y}) - \beta \cdot I(Z; \mathscr{X})$$

dove il primo termine misura la predittività della rappresentazione e il secondo la sua ridondanza rispetto all'input. Il parametro β controlla il compromesso tra completezza e compattezza. Per $\beta \to 0$ si ottiene la rappresentazione identità; per $\beta \to \infty$ si ottiene la statistica sufficiente minimale di $\mathscr{X}$ per predire $\mathscr{Y}$. Ogni world model corrisponde a una scelta specifica di $\mathscr{Y}$ e di β: nell'RSSM, $\mathscr{Y}$ è la traiettoria futura e β è il peso del termine KL nell'ELBO; nella JEPA, $\mathscr{Y}$ è la rappresentazione target e β è implicitamente determinato dalla scelta del masking e dalla normalizzazione della loss; nel CPC, $\mathscr{Y}$ è l'osservazione futura k passi avanti e β è codificato nell'architettura del predittore lineare.

Il riconoscimento di questa struttura comune ha implicazioni pratiche immediate. Suggerisce che i diversi paradigmi non sono in competizione ma lungo una curva di trade-off informazionale: la domanda non è quale sia il world model migliore in assoluto, bensì quale posizione su questa curva sia ottimale per il task specifico. La robotica di manipolazione fine, che richiede rappresentazioni ad alta risoluzione fisica per il controllo, tende verso β piccoli (rappresentazioni più complete); la pianificazione ad alto livello su sequenze linguistiche tende verso β grandi (rappresentazioni più astratte). Una gerarchia di world model, come quella proposta da LeCun, è interpretabile come una famiglia di soluzioni a diversi valori di β, organizzata verticalmente per livello di astrazione.

3. Il dualismo generativo/predittivo e la sua risoluzione parziale

La tensione più dibattuta nel campo dei world model — quella tra l'approccio generativo, che mira a modellare la distribuzione degli stati futuri nello spazio degli input, e l'approccio predittivo-latente della JEPA, che opera esclusivamente nello spazio delle rappresentazioni — si rivela, ad un'analisi formale più attenta, non una dicotomia assoluta bensì un continuum parametrizzato dalla scelta del decoder.

Per vederlo, si consideri un world model con encoder $f_\theta: \mathcal{O} \to \mathbb{R}^d$, modello di transizione $p_\varphi(z_{t+1} \mid z_t, a_t)$, e decoder $p_\psi(o_{t+1} \mid z_{t+1})$. La loss di training è:

$$\mathcal{L} = \mathbb{E}[-\log p_\psi(o_{t+1} \mid z_{t+1})] + \beta \cdot \mathrm{KL}[q_\varphi(z_{t+1} \mid o_{t+1}) \,\|\, p_\varphi(z_{t+1} \mid z_t, a_t)]$$

Quando il decoder p_ψ è una δ di Dirac sulla media del latente — cioè, quando $\mathrm{Dec}(z) = z$ e la loss di ricostruzione diventa $\|z_{t+1} - \hat{z}_{t+1}\|^2$ nello spazio latente — la loss coincide esattamente con quella di I-JEPA: si predice nello spazio latente, senza ricostruzione dell'osservazione. Quando invece il decoder è un modello di diffusione o un flusso normalizzante potente, il sistema produce campioni realistici nello spazio degli input, avvicinandosi al paradigma generativo completo. Il parametro che media tra i due estremi è la capacità espressiva del decoder: il continuum decoders triviali → decoders profondi → decoders generativi di stato dell'arte traccia esattamente la transizione dal paradigma JEPA puro al paradigma generativo puro.

Questa struttura unificante suggerisce che la domanda corretta non sia "JEPA o modello generativo?" bensì "quale capacità del decoder è giustificata dalla complessità del task?". Per task di controllo in bassa dimensione, dove la pianificazione richiede solo previsioni latenti accurate, un decoder triviale è sufficiente e il paradigma JEPA è ottimale. Per task di simulazione visiva realistica — generazione di video, addestramento di policy per robot in ambienti visualmente complessi — un decoder potente è necessario, e il paradigma generativo giustifica il suo costo computazionale addizionale. La JEPA con decoder di diffusione, discussa nel capitolo XVII, è la

posizione intermedia che massimizza l'efficienza dell'apprendimento delle rappresentazioni mantenendo la capacità di sintesi visiva quando necessario.

4. Il principio di invarianza strutturale come denominatore comune

Un secondo piano di convergenza, forse meno esplicito nel dibattito corrente ma teoricamente più profondo, riguarda il ruolo delle invarianze e delle simmetrie nell'architettura dei world model. Ogni approccio discusso in questa monografia incorpora, esplicitamente o implicitamente, un insieme di ipotesi sulle strutture del mondo che devono essere catturate e su quelle che possono essere scartate.

Nelle reti hamiltoniane, l'invarianza è quella della conservazione dell'energia: il campo vettoriale della dinamica è vincolato a derivare da una funzione scalare attraverso la struttura simplettica, garantendo per costruzione che le simulazioni rispettino le leggi di conservazione della meccanica classica. Nelle architetture equivarianti — EGNN, SE(3)-Transformer — l'invarianza è quella rispetto al gruppo euclideo: la rete garantisce che la previsione sia covariante rispetto a traslazioni e rotazioni, eliminando dall'apprendimento tutta la variabilità dovuta alla posizione assoluta nel riferimento inerziale. Nelle gerarchie di JEPA, l'invarianza è quella rispetto al masking: il predittore deve produrre rappresentazioni del target invarianti rispetto all'identità specifica della regione mascherata, catturando solo le strutture semanticamente stabili. Nei modelli causali di tipo CausalDreamer, l'invarianza è quella rispetto agli interventi: la fattorizzazione della distribuzione rispetta la struttura del DAG causale, garantendo che le previsioni siano corrette anche per configurazioni di stati prodotte da interventi non osservati nel training.

La struttura comune è questa: ogni world model di successo incorpora un prior strutturale sulla classe di invarianze che governa il dominio target. Un prior troppo debole porta all'overfitting alle correlazioni spurie del training set; un prior troppo forte impone vincoli che il dominio reale non rispetta, degradando la qualità delle previsioni sulle traiettorie al limite delle ipotesi vincolanti. Il disegno ottimale del prior strutturale è, in senso bayesiano, il cuore del problema di progettazione di un world model per un dominio specifico.

Formalizzando, sia $\mathcal{G}$ il gruppo di simmetrie del dominio fisico (o, più in generale, la struttura di invarianza rilevante, che può essere topologica piuttosto che di gruppo). Un world model ottimale per il dominio è una coppia (f_θ, p_φ) dove f_θ è equivariante rispetto a $\mathcal{G}$ e p_φ è invariante rispetto ai sottogruppi di $\mathcal{G}$ che lasciano invariante il reward. La condizione di equivarianza è:

$$f_\theta(g \cdot o) = \rho(g) \cdot f_\theta(o) \text{ per ogni } g \in \mathcal{G}$$

dove ρ è la rappresentazione di $\mathcal{G}$ nello spazio latente. Quando ρ è la rappresentazione banale (tutte le g agiscono come l'identità), f_θ è invariante; quando ρ è la rappresentazione aggiunta, f_θ è pienamente covariante. I world model che

incorporano questa struttura mostrano empiricamente un vantaggio di campionamento proporzionale alla dimensione del gruppo $\mathcal{G}$, poiché ogni traiettoria osservata fornisce $|\mathcal{G}|$ traiettorie "equivalenti" per l'apprendimento, anche se solo una è stata effettivamente raccolta.

5. La fenomenologia dell'emergenza e il problema della soglia

Un tema trasversale che emerge da quasi ogni capitolo di questa monografia — dalle leggi di scala dei grandi modelli fondazionali all'analisi topologica delle rappresentazioni, dalla fisica statistica dell'apprendimento profondo alle transizioni di fase nell'apprendimento auto-supervisionato — è quello dell'emergenza: la comparsa, al di sopra di certe soglie di scala o di training, di capacità qualitativamente nuove che non si possono prevedere interpolando il comportamento al di sotto della soglia.

Dal punto di vista teorico, l'emergenza nei world model può essere caratterizzata attraverso il linguaggio delle transizioni di fase. Si definisca la funzione di scala del world model $\Phi(C)$ come la performance media su un benchmark di pianificazione standardizzato, in funzione del budget computazionale C (numero di parametri × numero di token di training × numero di passi di ottimizzazione). Empiricamente, $\Phi(C)$ segue approssimativamente la legge:

$$\Phi(C) = \Phi_\infty - A \cdot C^{-\alpha}$$

per C al di sotto di una soglia critica C*, con esponente $\alpha > 0$ e asintoto Φ_∞. Al di sopra di C*, alcune capacità — comprensione della causalità fisica, generalizzazione compositiva, ragionamento controfattuale — mostrano una transizione quasi discontinua: $\Phi_{emergente}(C) \approx 0$ per $C < C^*$ e $\Phi_{emergente}(C) > 0$ per $C > C^*$, con una banda di transizione relativamente stretta rispetto alla scala logaritmica di C.

Il punto teoricamente cruciale, argomentato convincentemente da Schaeffer, Miranda e Koyejo (2023), è che molte emergenze apparenti dipendono dalla non-linearità della metrica di valutazione piuttosto che da una discontinuità nella distribuzione di probabilità sottostante. Tuttavia, alcune emergenze resistono a questa critica: in particolare, quelle legate alla ricombinazione compositiva di conoscenze distanti, dove la condizione di generalizzazione richiede la co-presenza di più strutture rappresentazionali che, al di sotto della soglia di scala, non coesistono nel modello. Queste emergenze genuine sono analoghe alle transizioni di fase del secondo ordine nella fisica dei materiali: la discontinuità nella derivata di Φ rispetto a log C segnala un cambiamento qualitativo nella struttura delle rappresentazioni, non semplicemente un cambiamento quantitativo nella loro qualità.

Per i world model, la domanda operativamente rilevante è: dove si trovano le soglie critiche C* per le capacità di comprensione fisica, di ragionamento causale, e di generalizzazione fuori distribuzione? I dati empirici disponibili suggeriscono che C* per la comprensione fisica qualitativa nei modelli visivi è dell'ordine di 10^{23}

FLOPs di training, mentre C* per il ragionamento causale strutturato appare molto più alta e forse non raggiungibile con i paradigmi attuali. Questa disomogeneità nelle soglie di emergenza implica che le architetture che incorporano strutture causali e fisiche esplicite — abbassando effettivamente C* attraverso l'injection di prior appropriati — mantengono un vantaggio sistematico rispetto ai modelli puramente data-driven, indipendentemente dalla scala di questi ultimi.

6. La convergenza epistemologica: dal rilevatore al costruttore

L'argomento più profondo per la convergenza dei paradigmi discussi in questa monografia non è di natura tecnica bensì epistemologica. Tutti gli approcci condividono, sia pure con declinazioni diverse, il rifiuto della concezione passivistica della percezione e dell'apprendimento che ha dominato l'intelligenza artificiale sino alla fine del XX secolo: la visione dell'agente come rilevatore di pattern, che registra le regolarità del mondo e le codifica passivamente in un sistema di predizione.

La concezione alternativa — incarnata nella biologia da Maturana e Varela con la nozione di autopoiesi, nella psicologia cognitiva da Piaget con quella di schema cognitivo, nella filosofia da Heidegger con quella di essere-nel-mondo, e nella teoria bayesiana della percezione da Friston con quella di cervello predittivo — vede l'agente come costruttore attivo di modelli del mondo, che impone struttura all'esperienza attraverso aspettative, azioni, e aggiornamenti guidati dall'errore di predizione. Un agente che costruisce un modello del mondo non registra ciò che vede; anticipa ciò che vedrà, e aggiorna il proprio modello in proporzione allo scarto tra aspettativa e realtà.

Questa convergenza epistemologica si manifesta tecnicamente nel ruolo centrale che il segnale di predizione gioca in tutti i paradigmi considerati. Nell'RSSM, il segnale di training è la discrepanza tra il prior (l'anticipazione) e il posterior (la correzione bayesiana). Nella JEPA, il segnale di training è la distanza tra la rappresentazione predetta del target e quella effettiva. Nei modelli di diffusione come world model, il segnale è l'errore di denoising — la discrepanza tra il rumore previsto e quello effettivamente aggiunto. Nel CPC, è la distanza contrastiva tra la previsione del futuro e i campioni negativi. In tutti i casi, l'apprendimento è guidato dall'errore di predizione: il segnale di apprendimento è identicamente zero se e solo se il modello anticipa perfettamente la struttura dell'esperienza.

Questa struttura unificante ha implicazioni normative per il progetto dei world model futuri. Suggerisce che il progresso nel campo non deriverà primariamente dall'aumento della capacità di rappresentazione (più parametri, più dati) bensì dal raffinamento della struttura della predizione: quale tipo di anticipazione il modello deve formare, a quale livello di astrazione, su quale orizzonte temporale, e con quale struttura dell'incertezza. La risposta a queste domande è, in ultima analisi, una risposta sulla natura della conoscenza che si vuole che il sistema acquisisca sul mondo.

7. Tensioni irrisolte e antinomie strutturali

Accanto alle convergenze, sarebbe intellettualmente disonesto tacere le tensioni irrisolte che attraversano il campo e che nessun paradigma attuale ha saputo dissolvere compiutamente.

La prima e più fondamentale è la tensione tra efficienza del campionamento e generalizzazione. I world model che generalizzano meglio fuori distribuzione — tipicamente quelli con prior strutturali forti, come le reti hamiltoniane o i modelli con fattorizzazione causale — richiedono che le ipotesi strutturali siano corrette, e degradano rapidamente quando il dominio viola quelle ipotesi. I world model che generalizzano in modo più robusto attraverso i domini — tipicamente i modelli di grande scala addestrati su corpora eterogenei — mostrano sistematici fallimenti di generalizzazione compositiva e causale che i prior strutturali avrebbero prevenuto. Il compromesso tra specificità del prior e robustezza della generalizzazione non ha una soluzione algoritmica soddisfacente nei paradigmi correnti, e rappresenta forse il limite teorico più importante del campo.

La seconda tensione riguarda il rapporto tra apprendimento online e stabilità della rappresentazione. Un world model utile per la pianificazione deve essere stabile: le rappresentazioni latenti non devono cambiare radicalmente tra un episodio e il successivo, altrimenti le politiche addestrate sulla base di quelle rappresentazioni diventano obsolete. Ma un world model capace di adattarsi a un mondo non stazionario deve aggiornare le proprie rappresentazioni quando l'ambiente cambia in modo sistematico. La tensione tra plasticità e stabilità — formalizzata dal fenomeno del catastrophic forgetting e dalla perdita di plasticità studiata da Nikishin e collaboratori — è la versione computazionale del dilemma biologico che il sistema ippocampale-corticale risolve attraverso la complementarietà dei sistemi di apprendimento rapido e lento. Nessuna architettura puramente parametrica ha risolto questo dilemma in modo soddisfacente, e le soluzioni più promettenti — come i replay buffer con priorità e la regolarizzazione EWC — ne mitigano gli effetti senza eliminarne le cause strutturali.

La terza tensione, forse la più profonda, è quella tra la descrizione distribuzionale del mondo — che è la forma naturale dell'apprendimento statistico — e la comprensione causale del mondo, che opera su meccanismi e non su correlazioni. Un world model addestrato su un corpus di transizioni con massima verosimiglianza apprende la distribuzione condizionale p(s_{t+1} | s_t, a_t) nella distribuzione di training; ma questa distribuzione cambia quando si interviene sul processo che genera s_t, perché interviene su variabili che sono anche cause degli stati futuri attraverso percorsi non mediati dall'azione a_t. Il do-calculus di Pearl formalizza precisamente la differenza tra la distribuzione osservazionale p(s_{t+1} | s_t, a_t) e la distribuzione interventistica p(s_{t+1} | do(a_t)): le due coincidono solo in assenza di confounders, condizione che raramente sussiste negli ambienti fisici reali. La risoluzione di questa tensione richiede un'integrazione genuina tra apprendimento statistico e struttura causale che rimane uno dei problemi aperti più importanti del campo.

8. Verso una teoria unificata: i principi fondativi

Pur con la cautela imposta dalla giovinezza del campo, è possibile identificare i principi fondativi che una teoria unificata dei world model dovrà articolare in modo coerente.

Il primo è il principio di predittività strutturata: la conoscenza di un sistema intelligente sul mondo si articola come anticipazione strutturata degli stati futuri, dove la struttura dell'anticipazione riflette le invarianze e le simmetrie del dominio. La bontà di un world model non si misura dalla fedeltà con cui riproduce le osservazioni passate, bensì dalla qualità con cui anticipa la struttura delle esperienze future in configurazioni mai incontrate durante l'addestramento.

Il secondo è il principio di compressione bayesiana: la rappresentazione ottimale degli stati del mondo è la statistica sufficiente minimale per le decisioni future, bilanciando la fedeltà all'esperienza sensoriale con la compattezza della codifica. Questo principio, variamente formulato come information bottleneck, energia libera variazionale, e MDL, costituisce il nucleo matematico dell'apprendimento delle rappresentazioni per i world model, e la sua violazione produce sistematicamente overfitting o underfitting rappresentazionale.

Il terzo è il principio di invarianza causale: un world model che generalizza genuinamente fuori distribuzione deve rappresentare i meccanismi causali del dominio piuttosto che le correlazioni statistiche che quei meccanismi producono nella distribuzione di training. La distinzione tra il livello associativo e il livello interventistico nella gerarchia causale di Pearl non è una sofisticazione teorica opzionale: è la condizione necessaria perché le previsioni del world model siano utili per la pianificazione in ambienti il cui comportamento è modificato dall'azione dell'agente.

Il quarto è il principio di grounding incarnato: la comprensione del mondo fisico e sociale richiede, almeno per gli agenti che devono agire in quel mondo, rappresentazioni ancorate all'esperienza sensorimotoria diretta. Le rappresentazioni puramente linguistiche, per quanto ricche di struttura semantica, mancano della calibrazione fisica che deriva dall'esperienza del feedback tattile, visivo, propriocettivo e cinestesico durante l'interazione con gli oggetti. Il programma della JEPA gerarchica per la robotica incarnata è, in questa prospettiva, non solo una scelta architetturale ma un impegno epistemologico sulla natura del grounding.

Il quinto è il principio di universalità scalare: le leggi di scala empiricamente osservate — la decrescita a potenza della loss in funzione del compute — riflettono proprietà universali della struttura statistica dell'informazione nel dominio, analoghe alle leggi universali della fisica critica. La comprensione teorica di queste leggi, attraverso la fisica statistica dell'apprendimento profondo, fornisce la base per prevedere il comportamento dei sistemi futuri e per ottimizzare l'allocazione delle risorse computazionali.

Questi cinque principi non sono indipendenti: il principio di predittività strutturata implica il principio di compressione bayesiana come strumento formale; il principio di invarianza causale rafforza il principio di predittività strutturata richiedendo che le strutture anticipate siano di natura meccanistica; il principio di grounding incarnato riempie di contenuto fisico il principio di invarianza causale; e il principio di universalità scalare fornisce la cornice quantitativa entro cui i primi quattro possono essere valutati empiricamente. Una teoria unificata dei world model articolerà questi principi in un sistema assiomatico coerente e li declinerà in architetture, algoritmi e metriche di valutazione operative.

9. Il ruolo unificante dell'architettura JEPA nella convergenza

È significativo che l'architettura attorno alla quale convergono più naturalmente i principi enunciati sia la JEPA gerarchica nella sua formulazione più ambiziosa. Essa incorpora la predittività strutturata come principio architetturale fondante (la previsione avviene nello spazio latente, non nello spazio degli input); realizza la compressione bayesiana attraverso il masking e la normalizzazione della loss; si presta naturalmente all'incorporazione di strutture causali attraverso la fattorizzazione delle variabili latenti; è concepita esplicitamente come architettura per agenti incarnati; e le sue leggi di scala sono state studiate empiricamente nella loro dipendenza dalla dimensione del modello e dalla quantità di dati.

La JEPA non è tuttavia una risposta completa alla domanda teorica che questa monografia ha cercato di sollevare. Le sue limitazioni — il rischio di collasso rappresentazionale, la difficoltà di garantire la completezza semantica delle rappresentazioni apprese, la necessità di strategie anti-collasso che non sono ancora completamente giustificate dalla teoria, l'assenza di un decoder generativo integrato per le applicazioni che richiedono simulazione visiva realistica — indicano che essa è un passo importante verso la teoria unificata, non la teoria stessa.

La teoria unificata che questo campo richiede dovrà essere in grado di derivare le architetture ottimali per classi di problemi specifici dai primi principi, di spiegare le emergenze osservate empiricamente come conseguenze di transizioni di fase nella struttura delle rappresentazioni, di fornire garanzie teoriche sulla generalizzazione fuori distribuzione in termini di distanza causale tra distribuzione di training e di test, e di connettere in modo formalmente preciso le rappresentazioni latenti delle reti neurali alle variabili causali che governano il dominio fisico. Questo è un programma ambizioso, certamente non completato nel corso di questa trattazione, e forse non completabile nell'arco di una singola generazione di ricercatori. Ma la convergenza dei paradigmi documentata in questa monografia indica che le fondamenta di tale teoria sono già state poste.

XX. Prospettive e Frontiere Aperte: L'Orizzonte dei Modelli del Mondo

1. L'inizio come fine: la domanda che rimane

Questa monografia ha preso avvio dalla nozione di modello interno formulata da Craik nel 1943 e si è conclusa — nel capitolo precedente — con la prospettiva di una teoria unificata dei world model. Il cerchio sembra chiudersi, ma in realtà si apre: ogni risposta che la ricerca degli ultimi anni ha saputo elaborare ha generato domande di maggiore profondità e generalità. L'obiettivo di questo capitolo finale non è dunque quello di tracciare un bilancio consolatorio, bensì quello di articolare con precisione le frontiere che rimangono aperte, di caratterizzare formalmente i problemi irrisolti, e di delineare le direzioni di ricerca che, sulla base della traiettoria teorica documentata, appaiono le più feconde per il decennio che si apre.

La domanda che rimane — e che probabilmente rimarrà, sotto forme sempre più raffinate, per molti anni — è la domanda sulla natura della comprensione. Non nel senso vago in cui il termine è usato nel discorso pubblico sull'intelligenza artificiale, ma nel senso tecnico e operativo introdotto nel capitolo sull'apprendimento causale: un sistema comprende il mondo fisico se è in grado di predire correttamente le conseguenze di interventi mai osservati durante l'addestramento, di ragionare correttamente su controfattuali rispetto alla propria esperienza passata, e di ricombinare in modo compositivamente corretto le strutture causali apprese in configurazioni radicalmente nuove. Questa definizione operativa della comprensione non è una dichiarazione di sufficienza — un sistema che la soddisfa potrebbe ancora mancare di molte proprietà che gli esseri umani associano alla comprensione genuina — ma è una condizione necessaria, e i sistemi attuali la soddisfano solo parzialmente e in modo fragile.

Due sviluppi recenti meritano di essere citati esplicitamente come progressi che iniziano a chiudere, almeno parzialmente, i gap qui identificati. Il primo è Rectified LpJEPA, discusso nel Capitolo V, che affronta il gap della sparsità controllabile nelle rappresentazioni JEPA: estendendo la distribuzione target dalla famiglia gaussiana isotropa a una famiglia più ampia (Generalized Gaussian Rettificata parametrizzata da p), il framework consente di regolare in modo principiato il grado di sparsità delle rappresentazioni latenti, avvicinando le codifiche apprese ai codici neurali sparsi biologicamente motivati e aprendo la strada a world model con rappresentazioni più interpretabili e più efficienti nella codifica dell'informazione sensoriale. Il secondo è C-JEPA, discusso nel Capitolo XII, che affronta il gap della struttura causale indotta dall'obiettivo di apprendimento: estendendo la predizione latente mascherata dal livello di patch al livello di oggetto, C-JEPA introduce implicitamente un bias causale inductive nell'obiettivo di training senza richiedere supervisione esplicita sulle relazioni causali, e la nozione di vicinato di influenza che ne emerge offre un'approssimazione operativa della struttura causale accessibile in spazi ad alta dimensionalità. Insieme, questi due filoni indicano che la distanza tra i principi architetturali della JEPA e i

requisiti di una comprensione fisica genuina si sta accorciando attraverso estensioni dell'obiettivo di predizione, non attraverso l'aggiunta di moduli simbolici separati.

2. Il problema della campionamento efficiente: il divario con la biologia

Il confronto tra l'efficienza del campionamento dei sistemi artificiali e quella degli organismi biologici rimane il problema aperto più quantificabile e forse più importante dell'intero campo. Un bambino di tre anni ha una comprensione del mondo fisico che nessun world model artificiale attuale raggiunge — comprende la permanenza degli oggetti, le relazioni causali elementari, le proprietà topologiche dello spazio, la distinzione tra agenti e oggetti — e l'ha acquisita attraverso un numero di esperienze sensorimotorie che è inferiore di diversi ordini di grandezza rispetto ai dataset di training dei sistemi più capaci.

Questa disparità non è una curiosità statistica: è il sintomo di una differenza fondamentale nella struttura dell'apprendimento. Gli organismi biologici portano all'esperienza un corpus di prior innati — codificati nel genoma attraverso miliardi di anni di selezione evolutiva in ambienti fisici — che funge da impalcatura per l'apprendimento esperienziale. Questi prior non sono semplici vincoli sulla forma delle rappresentazioni; sono strutture computazionali complete che implementano, fin dalla nascita, meccanismi di tracking degli oggetti, di inferenza causale, di attribuzione dell'agentività. La domanda che ne consegue è: come si possono incorporare prior analoghi nelle architetture artificiali senza rinunciare alla generalità che è la forza dei sistemi data-driven?

Una risposta parziale è quella già discussa con le architetture fisicamente informate: le reti hamiltoniane, le architetture equivarianti, i modelli con fattorizzazione causale, i world model con prior geometrici iperbolici. Ciascuno di questi approcci incorpora un subset specifico dei prior fisici che governano il mondo macroscopico. Ma nessuno di essi cattura la ricchezza e la flessibilità dei prior biologici, che si adattano dinamicamente alla struttura dell'esperienza e si riorganizzano in risposta a evidenze anomale — la cosiddetta violazione delle aspettative che nei neonati produce l'orientamento verso gli stimoli fisicamente impossibili.

Formalizzando il problema: sia $\mathscr{D}_n$ un dataset di n traiettorie campionate dalla distribuzione degli ambienti fisicamente plausibili. Sia $\varepsilon_n(M)$ l'errore di generalizzazione del world model M addestrato su $\mathscr{D}_n$, misurato sulla capacità di predire correttamente le conseguenze di interventi mai osservati. Il problema del campionamento efficiente chiede di trovare l'architettura M^* che minimizza $\varepsilon_n(M)$ per valori piccoli di n — idealmente paragonabili a quelli degli organismi biologici — soggetto al vincolo che M sia implementabile su hardware fisicamente realizzabile. Il bound di generalizzazione standard:

$$\varepsilon_n(M) \leq \varepsilon_\infty(M) + C \cdot \sqrt{(VC_dim(M) / n)}$$

suggerisce che minimizzare ε_n richiede di minimizzare la dimensione di Vapnik-Chervonenkis della classe di ipotesi di M — cioè di usare prior forti che restringano lo spazio delle soluzioni compatibili con l'esperienza. Ma minimizzare VC_dim rischia di vincolare la classe di ipotesi in modo troppo ristretto, aumentando ε_∞(M) — l'errore asintotico con dati infiniti. Il compromesso ottimale tra capacità e prior è la forma computazionale del problema della generalizzazione nei sistemi biologici.

3. Il problema del grounding fisico: dalla statistica alla causalità

Il secondo problema aperto fondamentale, già introdotto nel capitolo sull'apprendimento causale e nel capitolo sulle leggi di scala, è quello del grounding fisico: in che senso le rappresentazioni latenti di un world model corrispondono alle variabili fisiche che governano il dominio? Questa domanda non è retorica; ha una risposta formale precisa, e quella risposta è al momento insoddisfacente.

Si consideri un world model M con encoder f_θ: $\mathcal{O} \to \mathbb{R}^d$, addestrato su traiettorie prodotte da un sistema fisico con stato latente s $\in \mathcal{S}$ e dinamica f_{true}: $\mathcal{S} \times \mathcal{A} \to \mathcal{S}$. Diciamo che M è fisicamente grounded se esiste una mappa φ: $\mathbb{R}^d \to \mathcal{S}$ tale che:

φ(f_θ(o)) ≈ s per ogni coppia (o, s) nella distribuzione di training

e tale che φ commuta con la dinamica:

φ(f_θ(o_{t+1})) ≈ f_{true}(φ(f_θ(o_t)), a_t)

La prima condizione richiede che le rappresentazioni latenti codifichino correttamente lo stato fisico; la seconda richiede che la dinamica nello spazio latente rifletta la dinamica fisica reale. Queste due condizioni, prese insieme, garantiscono che la pianificazione nello spazio latente sia equivalente alla pianificazione nello spazio degli stati fisici.

Il problema è che l'esistenza di φ non è garantita dall'addestramento per massima verosimiglianza, né dal paradigma JEPA, né da alcun approccio puramente data-driven. L'encoder f_θ può apprendere una rappresentazione perfettamente predittiva sulla distribuzione di training senza che quella rappresentazione sia fisicamente interpretabile: potrebbe codificare combinazioni non lineari delle variabili fisiche, o addirittura variabili spurie che sono correlate con s nella distribuzione di training ma non in modo causalmente stabile.

Il teorema di identificabilità di Locatello e collaboratori — citato nel capitolo sul disentanglement — stabilisce che, senza supervisione aggiuntiva, la mappa φ non è identificabile in modo univoco: infinite famiglie di encoder diversi producono rappresentazioni latenti ugualmente predittive ma strutturalmente diverse. La soluzione richiede necessariamente informazione aggiuntiva: etichette supervisionate

(anche scarse), supervisione debole attraverso interventi sperimentali, o vincoli architetturali che implementino le simmetrie della fisica.

Le architetture fisicamente informate — reti hamiltoniane, architetture lagrangiane, modelli con struttura simplettica — forniscono vincoli architetturali che restringono la classe delle soluzioni compatibili con i dati, aumentando la probabilità che φ esista e sia identificabile. Ma questa soluzione funziona solo per i domini in cui le simmetrie fisiche rilevanti sono note a priori — come la meccanica classica o la meccanica quantistica — e fallisce per i domini complessi in cui le variabili causali non sono identificabili analiticamente.

La frontiera aperta è dunque quella di un framework generale per il grounding fisico che sia data-driven ma causalmente strutturato: un approccio che apprenda le variabili fisiche rilevanti dalla struttura delle traiettorie, senza richiedere la loro specifica a priori, ma garantendo che le rappresentazioni apprese soddisfino le condizioni di identificabilità causale. Il programma della causal representation learning di Schölkopf e collaboratori è la proposta più sistematica in questa direzione, ma rimane largamente aperto nelle sue implementazioni pratiche per scenari realisticamente complessi.

4. Il problema della composizionalità: dalla interpolazione alla generalizzazione strutturale

Il terzo grande problema aperto riguarda la composizionalità delle rappresentazioni dei world model: la capacità di ricombinare in modo corretto concetti, relazioni e dinamiche apprese separatamente per fare predizioni su configurazioni mai incontrate durante il training. Questo problema è distinto dalla generalizzazione fuori distribuzione in senso statistico: non si tratta di predire correttamente su variazioni quantitative rispetto alla distribuzione di training, bensì di ricombinare strutturalmente elementi diversi.

L'esempio paradigmatico, introdotto nel capitolo sull'apprendimento causale, è quello dell'oggetto con proprietà fisiche incrociate: un world model addestrato su sfere che rotolano e cubi che scivolano non generalizza correttamente, in generale, al comportamento di sfere che scivolano o di cubi che rotolano. La ragione è che il modello apprende correlazioni tra la forma dell'oggetto e il tipo di moto nella distribuzione di training, invece di apprendere le proprietà fisiche separate — le proprietà della forma (geometria del contatto) e le proprietà del moto (attrito, inerzia) — che determinano il moto in modo composizionale.

Formalizzando, si consideri una coppia di variabili latenti (z_forma, z_dinamica) dove z_forma codifica le proprietà geometriche e z_dinamica le proprietà dinamiche. La transizione di stato è:

$$s_{t+1} = f(s_t, a_t; z_{forma}, z_{dinamica})$$

con la fattorizzazione che riflette l'indipendenza causale tra le due classi di proprietà. Un world model che apprende questa fattorizzazione predice correttamente le

proprietà della sfera-scivolante come la combinazione f(s, a; z_sfera, z_scivolante), anche se questa combinazione specifica non è mai stata osservata. Un world model che non apprende la fattorizzazione — o che la apprende solo parzialmente — fallisce su questa inferenza.

Le architetture che promuovono esplicitamente la composizionalità — le slot-based representations, i modelli a grafo, il CausalDreamer con fattorizzazione delle variabili latenti — mostrano miglioramenti significativi rispetto alle architetture monolitiche sui benchmark di generalizzazione compositiva, ma il gap rispetto alla generalizzazione umana rimane ampio. Il problema aperto è di caratterizzare teoricamente le condizioni sotto le quali una rappresentazione latente è composizionalmente corretta — non solo predittivamente accurata sulla distribuzione di training — e di derivare architetture che soddisfino queste condizioni per costruzione piuttosto che per ottimizzazione.

Una congettura naturale, supportata dai risultati sull'IRM di Arjovsky e collaboratori, è che la composizionalità corretta sia equivalente all'invarianza causale delle rappresentazioni: una rappresentazione è composizionalmente corretta se e solo se i meccanismi causali che la governano sono stabili rispetto a interventi sulle variabili contestuali. Dimostrare formalmente questa equivalenza e derivarne algoritmi di addestramento pratici è una delle frontiere teoricamente più ricche del campo.

5. Il problema dell'efficienza del controllo: dalla comprensione all'azione

La quarta frontiera aperta riguarda il divario tra la qualità delle rappresentazioni apprese da un world model e la qualità della politica di controllo che ne deriva. Un world model può avere rappresentazioni eccellenti — catturare correttamente la fisica del dominio, generalizzare a configurazioni nuove, distinguere variabili causali da spurie — e tuttavia produrre politiche sub-ottimali, perché il bridge tra comprensione e azione non è ancora risolto.

Il problema è duplice. Da un lato, la pianificazione nel modello latente può divergere dalla pianificazione nell'ambiente reale a causa dell'errore di compounding: un piccolo errore di previsione ad ogni passo si amplifica lungo l'orizzonte di pianificazione, producendo piani che si rivelano disastrosi quando eseguiti nel mondo reale. Da un altro lato, le politiche ottimali nel modello possono sfruttare aspetti del modello che non corrispondono ad aspetti del mondo reale — il model exploitation discusso nella sezione su Dyna.

Una formalizzazione utile del problema di compounding usa il simulation lemma di Kakade e Langford (2002):

$$|V^*(s) - V^{\pi^*_{model}}(s)| \leq 2\gamma \, \varepsilon_{model} \, R_{max} / (1-\gamma)^2$$

dove V^* è il valore ottimale nell'ambiente reale, $V^{\pi^*_{model}}$ è il valore della politica ottimale nel modello, ε_{model} è la distanza distribuzionale (total variation) tra le transizioni del modello e quelle reali, R_{max} è la ricompensa massima, e γ il fattore di

sconto. Questo bound mostra che il divario tra la politica nel modello e quella nell'ambiente reale scala linearmente con l'errore del modello e quadraticamente con $1/(1-\gamma)$: per orizzonti lunghi ($\gamma \to 1$), anche piccoli errori del modello producono grandi perdite di performance.

La ricerca sugli algoritmi di training del mondo simulato ha prodotto diverse strategie per mitigare questo problema — orizzonti brevi, ensemble di modelli, calibrazione dell'incertezza, accesso periodico all'ambiente reale — ma nessuna le elimina strutturalmente. La frontiera aperta è quella di condizioni teoricamente garantite sotto le quali la pianificazione nel modello è sicura: condizioni che garantiscano che la politica ottimale nel modello sia anche ε-ottimale nell'ambiente reale, per qualsiasi $\varepsilon > 0$ specificato dall'utente, senza richiedere un numero di campioni reali che vanifica il vantaggio del model-based approach.

Una direzione promettente, ispirata dalla teoria del controllo robusto, è quella dei world model pessimistici: modelli che, invece di stimare il comportamento atteso del sistema, stimano il worst-case behavior entro un insieme di incertezza calibrato. La pianificazione su questi modelli produce politiche conservatrici che funzionano bene anche nel caso peggiore compatibile con i dati osservati, garantendo sicurezza senza richiedere la perfetta conoscenza del modello. Ma la calibrazione dell'insieme di incertezza — quanto conservativi essere, e su quali dimensioni dello spazio degli stati — è a sua volta un problema aperto.

6. Il problema dell'apprendimento continuo nella non-stazionarietà

La quinta frontiera aperta è quella dell'apprendimento continuo in ambienti non stazionari: la capacità di un world model di aggiornare le proprie rappresentazioni man mano che il dominio cambia, senza dimenticare le strutture dinamiche già apprese che rimangono valide.

Questa capacità è fondamentale per qualsiasi applicazione robotica reale: i robot operano in ambienti che cambiano nel tempo (nuovi oggetti vengono introdotti, la disposizione degli ostacoli varia, le proprietà dei materiali si modificano per usura o condizioni ambientali), e un world model che non si aggiorna diventa progressivamente meno utile per la pianificazione.

Il problema del catastrophic forgetting — la perdita drammatica di conoscenza precedente quando si addestra su nuovi dati — è ben compreso nella sua fenomenologia, ma non nella sua struttura causale profonda. La spiegazione più convincente, in termini di paesaggio di perdita, è quella di Nikishin e collaboratori: il training continuo su sequenze di task porta all'accumulo di neuroni morti (unità con attivazioni costantemente nulle) e alla degradazione della condizione numerica delle rappresentazioni, riducendo progressivamente la capacità del modello di apprendere strutture nuove — la perdita di plasticità — anche indipendentemente dal problema dell'oblio.

Il sistema biologico che risolve questo problema — la separazione tra memoria episodica hippocampale (apprendimento rapido, specifico) e memoria semantica corticale (apprendimento lento, generalizzante), con replay notturno per la consolidazione — suggerisce che la soluzione architettuale richieda, come minimo, una separazione strutturale tra i meccanismi di acquisizione rapida e quelli di consolidazione lenta. I replay buffer, la distillazione della conoscenza, e l'EWC sono implementazioni approssimate di questa separazione, ma nessuna cattura la dinamica adattiva con cui il sistema biologico decide quale conoscenza consolidare e quale mantenere come specifica episodica.

La frontiera formale è quella di un framework che garantisca la coesistenza di tre proprietà che in generale si escludono a vicenda — stabilità (preservare la conoscenza precedente), plasticità (acquisire conoscenza nuova), e generalizzazione (trasferire la conoscenza tra contesti) — e che derivi le condizioni sotto le quali tutte e tre possono essere soddisfatte simultaneamente con perdite accettabili. Il legame teorico con la geometria dello spazio di Fisher e con la teoria PAC-Bayes suggerisce che queste condizioni abbiano a che fare con la struttura di curvatura del paesaggio di perdita nella regione dei parametri correnti, ma la formalizzazione precisa è ancora incompleta.

7. Il problema dell'allineamento tra obiettivi e comportamento nei sistemi autonomi

La sesta frontiera aperta, di natura parzialmente diversa dalle precedenti perché si situa all'intersezione tra teoria tecnica e riflessione normativa, riguarda l'allineamento tra gli obiettivi impliciti appresi da un world model attraverso il reinforcement learning e gli obiettivi espliciti che gli esseri umani intendono specificare. Questa frontiera è particolarmente rilevante per i world model autonomi — sistemi che pianificano su orizzonti lunghi e eseguono sequenze di azioni senza supervisione umana continua.

Il problema dell'allineamento nei world model ha una struttura diversa da quello dei modelli linguistici. Per questi ultimi, l'allineamento riguarda primariamente la calibrazione delle preferenze del modello rispetto alle preferenze umane espresse attraverso feedback linguistico. Per i world model autonomi, l'allineamento riguarda la compatibilità tra la funzione di ricompensa che il sistema ottimizza — tipicamente un proxy matematico dell'obiettivo desiderato — e il reale obiettivo umano, che è spesso incompletamente specificabile in forma matematica.

La reward misspecification — la discrepanza tra la funzione di ricompensa proxy e l'obiettivo desiderato — produce sistematicamente comportamenti indesiderati quando il sistema opera in configurazioni lontane dalla distribuzione di training: trova soluzioni che massimizzano il proxy senza soddisfare l'obiettivo. Questo fenomeno, studiato formalmente da Krakovna e collaboratori (2020) e da Hadfield-Menell e collaboratori (2016, 2017), è esacerbato nei world model perché la pianificazione a lungo orizzonte amplifica le piccole discrepanze tra proxy e obiettivo in comportamenti macroscopicamente indesiderati.

La soluzione più promettente, in termini tecnici, è l'apprendimento inverso della ricompensa da dimostrazioni — usando le traiettorie di esperti umani per inferire la funzione di ricompensa implicita piuttosto che specificarla esplicitamente. Ma l'IRL standard assume che le dimostrazioni siano ottimali rispetto all'obiettivo latente, ipotesi sistematicamente violata nella pratica. Le estensioni bayesiane dell'IRL, che modellano esplicitamente la sub-ottimalità del dimostratore, e i metodi di cooperative IRL di Hadfield-Menell, che modellano l'interazione tra umano e sistema come un gioco cooperativo sotto incertezza sulla funzione di ricompensa, sono le direzioni più promettenti, ma rimangono largamente aperte nelle loro implicazioni teoriche per i world model con orizzonti di pianificazione lunghi.

8. Il problema dell'integrazione con il mondo sociale

La settima frontiera aperta è quella del world model come strumento di comprensione del mondo sociale — non solo fisico. I capitoli su RLHF e sui modelli di linguaggio come world model hanno già tracciato una connessione tra la comprensione linguistica e la simulazione dei comportamenti sociali. Ma la comprensione del mondo sociale richiede capacità che vanno oltre la modellazione statistica delle distribuzioni di comportamenti osservati: richiede la rappresentazione degli stati interni degli agenti — credenze, desideri, intenzioni, aspettative — e la capacità di predire come questi stati cambiano in risposta alle azioni proprie e altrui.

Questa capacità — la Teoria della Mente computazionale, introdotta nel capitolo sui LLM come world model — è necessaria per qualsiasi robot che debba operare in ambienti condivisi con esseri umani: deve anticipare le intenzioni del collaboratore umano, adattare le proprie azioni alle sue aspettative, e comunicare in modo efficiente i propri piani. I benchmark attuali — Task of Mind, False Belief tasks — mostrano che i sistemi più grandi raggiungono prestazioni elevate su questi test, ma i test stessi sono criticamente vulnerabili alla critica di Ullman: possono essere risolti attraverso pattern superficiali senza richiedere una genuina rappresentazione degli stati mentali.

La frontiera formale è quella di una ToM computazionale che sia:

(i) basata su un modello esplicito degli stati mentali degli agenti come distribuzioni di probabilità su stati del mondo fisico e sociale — cioè, un POMDP ricorsivo dove lo stato di ogni agente include credenze sullo stato del mondo e sugli stati degli altri agenti;

(ii) aggiornabile in tempo reale attraverso l'osservazione del comportamento degli altri agenti;

(iii) utilizzabile per la pianificazione in ambienti multi-agente, dove le azioni di un agente influenzano lo stato interno degli altri agenti che a loro volta modificano le loro azioni future.

Questo problema — la ToM come inferenza bayesiana ricorsiva su POMDP annidati — è computazionalmente intrattabile nella sua formulazione esatta, e le approssimazioni note (modelli di agente fisso, approssimazioni mean-field, inferenza variazionale) perdono tipicamente le dipendenze ricorsive che sono il nucleo del ragionamento sociale. Lo sviluppo di approssimazioni scalabili che preservino la struttura ricorsiva è una delle sfide teoriche più difficili e più importanti della prossima decade.

9. Il problema della robustezza epistemica e dell'auto-conoscenza del limite

Un'ultima frontiera, forse la meno formalizzata ma non meno importante, è quella della robustezza epistemica dei world model: la capacità di un sistema di riconoscere i limiti della propria conoscenza e di agire con cautela quando opera in regioni dello spazio degli stati o delle situazioni che sono fuori dalla propria competenza.

Questa capacità — che nella letteratura sull'apprendimento automatico viene spesso denominata "calibrazione" o "out-of-distribution detection" — è banalmente importante per la sicurezza dei sistemi robotici autonomi: un robot che non sa di non sapere qualcosa è più pericoloso di uno che conosce i propri limiti e li rispetta. Ma formalizzarla in modo rigoroso è tutt'altro che banale.

La distinzione tra incertezza epistemica e aleatoria, introdotta nel capitolo sui POMDP, fornisce una base formale: l'incertezza epistemica è quella riducibile con più dati, e un sistema ben calibrato dovrebbe segnalarla con alta entropia nelle proprie previsioni nelle regioni poco esplorate dello spazio degli stati. Ma questa distinzione è operativamente difficile da implementare nei modelli neurali profondi: la varianza dell'ensemble misura approssimativamente l'incertezza epistemica, ma la dimensione degli ensemble necessari per una stima affidabile cresce rapidamente con la dimensione del dominio.

Le metriche di conformalità — che costruiscono insiemi di predizione con copertura garantita senza ipotesi distribuzionali forti — offrono un'alternativa teoricamente solida. Per un world model M, si definisce l'insieme di predizione conforme al livello α come:

$$C_{n,\alpha}(s_t, a_t) = \{s : \text{score}(s, M(s_t, a_t)) \leq q_{\lceil (1-\alpha)(n+1) \rceil}(S_{\text{calibration}})\}$$

dove score è una funzione di non-conformità (ad esempio la distanza tra lo stato predetto e quello osservato) e q è il quantile appropriato calcolato su un insieme di calibrazione. Sotto ipotesi di scambiabilità dei dati — una condizione debole soddisfatta da molti scenari di interesse pratico — $C_{n,\alpha}$ ha garanzia di copertura:

$$\mathbb{P}(s_{t+1} \in C_{n,\alpha}(s_t, a_t)) \geq 1 - \alpha$$

indipendentemente dalla distribuzione dei dati. Questa garanzia di copertura è esattamente ciò che serve per la pianificazione robusta: un sistema che sa che la

traiettoria futura reale sarà con probabilità $1-\alpha$ contenuta nel suo insieme di predizione può pianificare tenendo esplicitamente conto di questo insieme.

La frontiera aperta è quella di estendere la conformalità alla pianificazione a lungo orizzonte — dove gli insiemi di predizione si propagano e si amplificano attraverso i passi del rollout — e di integrare questo framework con le architetture di world model esistenti in modo computazionalmente trattabile.

10. Una visione di sintesi

Dalla prospettiva di questo capitolo finale, il problema dei world model appare non come un problema tecnico isolato bensì come il cuore del più ampio problema dell'intelligenza artificiale genuina. La comprensione del mondo — fisica, causale, sociale — è la condizione necessaria per qualsiasi forma di intelligenza che aspiri a essere autonoma, robusta e sicura nel mondo reale. Un sistema che non capisce il mondo fisico non può pianificare in modo affidabile; uno che non capisce il mondo causale non può prevedere le conseguenze delle proprie azioni; uno che non capisce il mondo sociale non può interagire con gli esseri umani in modo sicuro ed efficace.

Il paradigma dei world model — nella sua articolazione come architettura di predizione latente gerarchica, arricchita da strutture causali, simmetrie fisiche, meccanismi di memoria, e grounding multimodale — è la risposta computazionale più sistematicamente sviluppata a questo problema. Non è ancora una risposta completa: le frontiere aperte enumerate in questo capitolo indicano che la teoria è ancora in via di costruzione, che le architetture operative ne realizzano solo approssimazioni, e che i benchmark disponibili non sono ancora in grado di discriminare la comprensione genuina dall'interpolazione sofisticata.

Ma la traiettoria del campo negli ultimi dieci anni suggerisce che il progresso sia rapido e le convergenze teoriche già sufficientemente profonde da consentire un ottimismo temperato. Il principio della predittività strutturata, l'asse variazionale come struttura unificante, la necessità del grounding incarnato, la promessa delle gerarchie multi-scala — questi elementi convergono verso un'architettura concettuale che non era disponibile nel 2015 e che non sarà ancora completa nel 2030. Il decennio che si apre sarà probabilmente quello in cui le fondamenta teoriche diventeranno sufficientemente solide da consentire la costruzione di sistemi con genuine capacità di comprensione del mondo fisico, causale e sociale.

Il traguardo è ambizioso ma non arbitrario. La sua realizzazione richiederà matematica più profonda, esperimenti più sistematici, e soprattutto una volontà di integrare tradizioni disciplinari che il sistema accademico tende a mantenere separate: la fisica statistica e l'ottimizzazione, la geometria differenziale e l'apprendimento automatico, la teoria causale e le reti neurali, la robotica incarnata e la comprensione del linguaggio. Questa monografia ha cercato di essere un contributo a tale integrazione — non perché abbia completato il lavoro, ma perché ha cercato di dimostrare che il lavoro è possibile e che gli strumenti per compierlo esistono già, sebbene sparpagliati tra

discipline che devono ancora imparare a riconoscersi come parti di un unico progetto intellettuale.

La comprensione del mondo, in ultima analisi, è la comprensione delle strutture di invarianza che rendono il mondo stabile, prevedibile, e navigabile. Quelle strutture esistono — la fisica le ha catalogate, la matematica le ha formalizzate, la biologia le ha implementate in miliardi di anni di evoluzione. Il compito che resta è insegnarle ai sistemi artificiali in modo tale che questi possano abitare il mondo con la stessa naturalezza con cui lo abitano gli agenti biologici che in quel mondo sono cresciuti.

XXI. Apprendimento per rinforzo model-based offline: MOPO, MOReL, MOBILE e la teoria della generalizzazione conservativa

1. La rottura epistemica del paradigma online: motivazione e collocazione

Il problema dell'apprendimento per rinforzo online ha per decenni dominato tanto la riflessione teorica quanto le applicazioni sperimentali del paradigma model-based: un agente interagisce ripetutamente con un ambiente, raccoglie transizioni, aggiorna il proprio modello del mondo e ottimizza una politica che può essere immediatamente eseguita e valutata. L'architettura di Dreamer, le varianti di MuZero e i sistemi di controllo robotico costruiti su RSSM condividono tutti questa ipotesi fondamentale — che il ciclo percezione-predizione-azione-raccolta di dati resti aperto e iterabile durante l'addestramento.

Esistono tuttavia vastissimi domini nei quali questa ipotesi è radicalmente inapplicabile. La medicina di precisione dispone di archivi di cartelle cliniche e traiettorie terapeutiche, ma non può permettersi cicli di esplorazione che mettano a rischio pazienti reali. L'ingegneria aeronautica conserva registrazioni di missioni e incidenti, ma non può esplorare politiche di controllo sperimentali su velivoli in volo. La guida autonoma genera ogni giorno petabyte di dati provenienti da flotte di veicoli, ma non autorizza il sistema ad apprendere mediante tentativi non supervisionati sul traffico reale. In questi contesti l'agente dispone di un corpus fisso di esperienze passate — un insieme di transizioni (s, a, r, s') raccolte da politiche comportamentali di natura eterogenea e arbitraria — e deve estrarre da esso una politica tanto efficace quanto possibile, senza alcuna possibilità di interagire nuovamente con l'ambiente.

Questa impostazione prende il nome di apprendimento per rinforzo offline (o, in parte della letteratura, *batch reinforcement learning*). La difficoltà principale che la contraddistingue non è di natura computazionale ma epistemica: il dataset offline copre soltanto una porzione ridotta dello spazio stati-azioni, e qualsiasi stima del valore di coppie (stato, azione) non osservate nella distribuzione dei dati è intrinsecamente affetta da un'incertezza non misurabile con i soli strumenti della

statistica frequentista standard. Tentare di massimizzare la politica su tali stime porta a un fenomeno caratteristico, noto come *sfruttamento dell'errore del modello* (*model exploitation* o *value overestimation*), nel quale l'ottimizzatore identifica coppie fuori distribuzione sulle quali la rete di valore assegna punteggi elevati per ragioni puramente estrapolative, senza che tali valori corrispondano a rendimenti reali.

Il contributo specifico del paradigma model-based all'offline RL non consiste in una semplice surrogazione del mondo reale con un simulatore appreso — operazione che riprodurrebbe e amplificherebbe esattamente il problema — bensì nell'introduzione esplicita di un meccanismo di penalizzazione dell'incertezza del modello che implementi una forma di pessimismo epistemico. Questo capitolo ricostruisce con rigore la teoria di tale pessimismo, ne deriva le garanzie teoriche quantitative, e ne analizza le tre principali incarnazioni algoritmiche: MOPO (Yu et al., 2020), MOReL (Kidambi et al., 2020) e MOBILE (Sun et al., 2023).

2. Il framework dell'offline RL: definizioni e struttura del problema

Si consideri un processo decisionale markoviano (MDP) $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma, \rho_0)$ dove $\mathcal{S}$ è lo spazio degli stati, $\mathcal{A}$ lo spazio delle azioni (si lavora inizialmente in spazi continui), $P: \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ la funzione di transizione, $r: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ la funzione di ricompensa, $\gamma \in (0,1)$ il fattore di sconto, e ρ_0 la distribuzione iniziale degli stati.

Il valore di una politica $\pi: \mathcal{S} \to \Delta(\mathcal{A})$ è definito come il rendimento scontato atteso:

$$J(\pi) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \mid s_0 \sim \rho_0, a_t \sim \pi(\cdot|s_t), s_{t+1} \sim P(\cdot|s_t, a_t)\right]$$

La funzione di valore $Q^\pi: \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ soddisfa l'equazione di Bellman:

$$Q^\pi(s,a) = r(s,a) + \gamma\, \mathbb{E}_{s' \sim P(\cdot|s,a),\, a' \sim \pi(\cdot|s')}[Q^\pi(s', a')]$$

e la politica ottimale π^* è definita da $\pi^*(s) = \operatorname{argmax}_a Q^*(s,a)$ dove $Q^* = Q^{\pi^*}$.

Nel setting offline, l'agente ha accesso a un dataset $D = \{(s_i, a_i, r_i, s'_i)\}_{i=1}^N$ raccolto da una politica comportamentale $\beta: \mathcal{S} \to \Delta(\mathcal{A})$ che è sconosciuta all'agente. Si definisce la distribuzione dei dati come $d^\beta(s,a)$, che rappresenta la distribuzione marginale stazionaria delle coppie stato-azione sotto β nel MDP M. La concentrabilità del dataset è caratterizzata dal coefficiente:

$$C^\pi = \max_{(s,a):\, d^\pi(s,a) > 0} \left[d^\pi(s,a) / d^\beta(s,a)\right]$$

dove d^π è la distribuzione stazionaria sotto la politica π da valutare. Quando C^π è piccolo, la politica π visitava prevalentemente coppie ben rappresentate nel dataset; quando è grande, π induce distribuzione su coppie sottorappresentate o assenti. Il problema di distributional shift si manifesta esattamente quando $C^\pi \gg 1$ per la politica che si desidera ottimizzare.

Un risultato classico (Kakade e Langford, 2002; Szepesvári, 2010) stabilisce che per ogni metodo di apprendimento per rinforzo offline, il bound sull'errore di stima del valore cresce almeno linearmente in C^π e inversamente nella radice della dimensione del dataset. Questo rende impossibile, in linea di principio, garantire la qualità di una politica che visiti coppie fuori distribuzione.

3. Il principio del pessimismo nell'offline RL: fondamenti teorici

La risposta principale a questo problema di trattabilità è il *principio del pessimismo sotto l'incertezza* (pessimism under uncertainty, PUU), il cui razionale è il seguente: se non si possono stimare affidabilmente i valori fuori distribuzione, conviene costruire un MDP modificato M̂ nel quale le regioni di bassa copertura dei dati ricevono penalità esplicite che disincentivano la politica dall'avventurarsi in esse. Formalmente, si definisce un MDP pessimistico M̂ = (𝒮, 𝒜, P̂, r̂, γ, ρ₀) dove il modello di transizione P̂ e la funzione di ricompensa r̂ sono appresi dai dati e corretti da un termine di penalità che dipende dall'incertezza del modello.

Il risultato teorico fondamentale che giustifica questo approccio è il seguente. Sia π̂ la politica ottimale su M̂ e sia u(s,a) una misura di incertezza del modello tale che, con alta probabilità, per ogni (s,a):

$$\| P(\cdot|s,a) - \hat{P}(\cdot|s,a) \|_1 \leq u(s,a)$$

allora, per ogni funzione di valore f limitata da V_max, vale:

$$| V^{\pi}(s) - \hat{V}^{\pi}(s) | \leq \gamma/(1-\gamma) \cdot \mathbb{E}_{\tau \sim \pi}[\Sigma_t \gamma^t u(s_t, a_t)] \cdot V_{max}$$

dove V̂^π è la stima del valore ottenuta sul modello appreso P̂. Questo bound si propaga lungo l'intera traiettoria: se la politica π visita frequentemente coppie con alta incertezza u, l'errore di stima del valore può diventare arbitrariamente grande.

La costruzione pessimistica risolve questo problema costruendo un MDP nel quale la ricompensa modificata è:

$$\hat{r}_{pessimistico}(s,a) = r(s,a) - \lambda \cdot u(s,a)$$

Il parametro $\lambda > 0$ controlla l'intensità della penalità. Ottimizzare π sul MDP modificato forza la politica verso regioni di bassa incertezza, cioè verso regioni ben coperte dai dati, riducendo il potenziale di sfruttamento estrapolativo del modello.

Il legame tra il bound sull'errore di stima del valore e la scelta ottimale di λ emerge da un'analisi di ottimizzazione minmax: si cerca il λ che minimizza il gap tra la performance della politica appresa nel MDP pessimistico e quella della migliore politica raggiungibile con i dati disponibili. Questo gap si chiama *suboptimality gap* ed è la grandezza chiave delle garanzie teoriche dell'offline MBRL.

4. MOPO: Model-based Offline Policy Optimization

MOPO (Yu et al., 2020) è il primo algoritmo offline model-based a formalizzare rigorosamente il principio del pessimismo e a derivarne garanzie quantitative di suboptimalità. La sua architettura combina un ensemble di reti neurali per la stima dell'incertezza del modello con una variante di Soft Actor-Critic (SAC) ottimizzata sull'MDP penalizzato.

4.1 Stima dell'incertezza tramite ensemble

Un ensemble di N modelli del mondo $\{\hat{P}_{\theta_i}\}_{i=1}^{N}$ viene addestrato per massima verosimiglianza sul dataset D. Ogni modello parametrizza una gaussiana diagonale sullo stato successivo e sulla ricompensa:

$$\hat{P}_{\theta_i}(s', r \mid s, a) = \mathcal{N}([s', r]; \mu_{\theta_i}(s,a), \mathrm{diag}(\sigma^2_{\theta_i}(s,a)))$$

La misura di incertezza epistemica adottata da MOPO è la varianza dell'ensemble sulle previsioni di stato:

$$u_E(s,a) = \max_i \| \mu_{\theta_i}(s,a) - \bar{\mu}(s,a) \|_2$$

dove $\bar{\mu}(s,a) = (1/N) \sum_i \mu_{\theta_i}(s,a)$ è la media dell'ensemble. Questa scelta è motivata dalla sua semplicità computazionale e dalla sua correlazione empirica con l'errore di predizione del modello sulle coppie fuori distribuzione. Una variante teoricamente più appropriata, discussa da Yu et al., utilizza la varianza massimale dell'ensemble:

$$u_V(s,a) = \max_i \| \sigma^2_{\theta_i}(s,a) \|_1$$

che misura l'incertezza aletoristica massima all'interno dell'ensemble.

Il dataset sintetico usato per l'ottimizzazione della politica viene generato mediante rollout del modello ensemble di lunghezza H a partire da stati campionati dal dataset reale D. La ricompensa di ogni transizione sintetica $(\hat{s}, \hat{a}, \hat{r}, \hat{s}')$ è modificata sottraendo la penalità:

$$\hat{r}_{mod}(\hat{s}, \hat{a}) = \hat{r}(\hat{s}, \hat{a}) - \lambda \cdot u_E(\hat{s}, \hat{a})$$

Il dataset composito $D_{composito} = D \cup D_{sintetico}$ viene poi utilizzato per l'addestramento di SAC, con la proporzione tra dati reali e sintetici come iperparametro.

4.2 Derivazione del bound di subottimalità

Il risultato teorico principale di MOPO è il seguente teorema. Sia $M = (\mathcal{S}, \mathcal{A}, P, r, \gamma)$ l'MDP reale, $M_\lambda = (\mathcal{S}, \mathcal{A}, \hat{P}, r - \lambda u, \gamma)$ l'MDP penalizzato, e $\hat{\pi}$ la politica ottimale su M_λ. Sia ε_m l'errore massimo di predizione del modello in norma L1 e ε_r l'errore

massimo sulla ricompensa, entrambi verificati con probabilità 1-δ attraverso un bound di concentrazione. Allora, con probabilità 1-δ:

$$J(\pi^*) - J(\hat{\pi}) \leq [2\gamma/(1-\gamma)^2] \cdot \sqrt{(2 \cdot C^{\hat{\pi}} \cdot |\mathcal{S}| \cdot |\mathcal{A}| / N \cdot \log(2 \cdot |\mathcal{S}| \cdot |\mathcal{A}| \cdot N/\delta))} \cdot V_{max}$$

Per derivare questo risultato si parte dalla decomposizione del gap di performance via il performance difference lemma di Kakade e Langford (2002):

$$J(\pi^*) - J(\hat{\pi}) = [1/(1-\gamma)] \cdot \mathbb{E}_{(s,a) \sim d^{\pi^*}}[Q^{\hat{\pi}}(s,a) - Q^{\hat{\pi}}(s, \hat{\pi}(s))]$$

dove d^{π^*} è la distribuzione stazionaria di π^*. Il termine dentro l'aspettazione misura il vantaggio di π^* su $\hat{\pi}$ a ogni passo. Questo si maggiora utilizzando l'equazione di Bellman simulata:

$$Q^{\hat{\pi}}(s,a) - V^{\hat{\pi}}(s) \leq r(s,a) + \gamma \, \mathbb{E}_{s' \sim P}[V^{\hat{\pi}}(s')] - V^{\hat{\pi}}(s) = r(s,a) + \gamma \, \mathbb{E}_{s' \sim P}[V^{\hat{\pi}}(s')] - r(s,\hat{\pi}(s)) - \gamma \, \mathbb{E}_{s'' \sim P}[V^{\hat{\pi}}(s'')] \leq 2 \, V_{max} \cdot |P(\cdot|s,a) - \hat{P}(\cdot|s,a)|_1 + \lambda \cdot u(s,a) + \varepsilon_r$$

Il termine $|P(\cdot|s,a) - \hat{P}(\cdot|s,a)|_1$ viene maggiorato tramite il bound di concentrazione sull'errore del modello, che per N transizioni osservate per la coppia (s,a) scala come $O(\sqrt{(|\mathcal{S}|/N)})$. Integrando sulla distribuzione d^{π^*} e applicando la definizione di coefficiente di concentrabilità $C^{\hat{\pi}}$ si ottiene il bound finale.

La scelta ottimale di λ emerge bilanciando il termine di errore del modello (che cresce quando si riduce λ, perché la politica visita regioni fuori distribuzione) con il termine di subottimalità indotto dalla penalizzazione eccessiva (che cresce con λ). L'analisi mostra che $\lambda^* = O(\sqrt{(|\mathcal{S}| / N)})$ è il valore che minimizza il bound.

4.3 Connessione con la teoria dell'informazione

Un'interpretazione alternativa del meccanismo di penalità di MOPO emerge dalla prospettiva dell'informazione mutua. Si consideri la misura di Fisher:

$$I_N(s,a) = N \cdot d^{\beta}(s,a)$$

che rappresenta la quantità di informazione disponibile nel dataset per la coppia (s,a). Quando $I_N(s,a)$ è piccola, l'incertezza epistemica u(s,a) è alta. La penalità $\lambda \cdot u(s,a)$ può essere quindi interpretata come un costo d'informazione: accedere a regioni di bassa copertura ha un costo proporzionale all'incertezza dell'informazione disponibile. Questa lettura ricollega MOPO al principio della minima sorpresa (*principle of minimum surprise*) di Friston, discusso nel capitolo sulla pianificazione bayesiana: un agente razionale dovrebbe preferire stati che non lo sorprendono, cioè stati la cui transizione è ben predicibile dal modello appreso.

5. MOReL: Model-based Offline RL e l'MDP pessimistico tramite stato di arresto

MOReL (Kidambi et al., 2020) propone un approccio concettualmente diverso da MOPO pur condividendone la filosofia pessimistica. Invece di modificare la funzione di ricompensa con un termine di penalità continuo, MOReL costruisce un nuovo MDP — denominato U-MDP (Unknown-penalized MDP) — nel quale le regioni di alta incertezza sono esplicitamente rappresentate come uno stato terminale assorbente con ricompensa altamente negativa.

5.1 Costruzione dell'U-MDP

Si addestra un ensemble di modelli $\{\hat{P}_{\theta_i}\}$ come in MOPO. Si definisce un discriminatore di distribuzione tramite la soglia $\varkappa > 0$:

$$\text{KNOWN}(s,a) = \mathbb{1}[u(s,a) \leq \varkappa]$$

dove u è una misura di incertezza dell'ensemble. La partizione dello spazio stati-azioni in regioni note (KNOWN) e ignote (UNKNOWN) è il nucleo concettuale di MOReL.

L'U-MDP è definito come $\tilde{M} = (\mathcal{S}, \mathcal{A}, \tilde{P}, \tilde{r}, \gamma, \rho_0)$ dove $\mathcal{S} = \mathcal{S} \cup \{s_halt\}$ con s_halt stato assorbente, e la funzione di transizione è:

$$\tilde{P}(s' \mid s,a) = \hat{P}(s' \mid s,a) \text{ se KNOWN}(s,a) = 1 \quad \tilde{P}(s_halt \mid s,a) = 1 \text{ se KNOWN}(s,a) = 0$$

La funzione di ricompensa è:

$$\tilde{r}(s,a) = r(s,a) \text{ se } s \neq s_halt \quad \tilde{r}(s_halt, a) = -\varkappa_r/(1-\gamma) \text{ con } \varkappa_r \geq 0$$

Il valore $-\varkappa_r/(1-\gamma)$ viene scelto in modo che il ritorno scontato atteso dallo stato di arresto sia $-\varkappa_r$, ovvero il peggio possibile scalato al denominatore delle serie geometriche. Qualsiasi politica che transita spesso nello stato di arresto riceve un valore molto negativo, il che la penalizza efficacemente.

La politica $\hat{\pi}$ ottimale sull'U-MDP viene trovata con qualsiasi algoritmo di RL standard (MOReL usa NPG, Natural Policy Gradient, nel paper originale). Poiché il modello è deterministico su KNOWN e abortisce su UNKNOWN, l'ottimizzazione è vincolata implicitamente alla regione ben coperta dai dati.

5.2 Garanzie teoriche di MOReL

Il teorema principale di MOReL afferma che, sotto opportune ipotesi di regolarità sulla funzione di valore e sul modello del mondo, la politica $\hat{\pi}$ ottenuta da MOReL soddisfa:

$$J(\pi^*) - J(\hat{\pi}) \leq \varepsilon_m \cdot V_{max} \cdot [\gamma/(1-\gamma)^2] + 2\varkappa_r \cdot P_{\{UNKNOWN\}}(\pi^*) \,/\, (1-\gamma)$$

dove ε_m è l'errore massimo di predizione del modello su KNOWN(s,a) = 1 e $P_{\{UNKNOWN\}}(\pi^*)$ è la probabilità che la politica ottimale π^* visiti regioni UNKNOWN.

Il secondo termine è particolarmente significativo: se la politica ottimale π^* non visita mai regioni UNKNOWN — cioè se il dataset copre tutte le coppie che π^* visiterebbe — allora il gap di suboptimalità è interamente controllato dall'errore del modello ε_m sulla regione KNOWN, che scala come $O(1/\sqrt{N})$. Questo è il caso favorevole in cui il dataset β è detto *ε_m-accurato* rispetto a π^*.

La differenza concettuale tra il bound di MOReL e quello di MOPO è la seguente. MOPO produce un bound che dipende esplicitamente dal coefficiente di concentrabilità $C^{\hat{\pi}}$, che può essere molto grande se la politica appresa $\hat{\pi}$ si avventura fuori dalla distribuzione dei dati. MOReL invece costruisce strutturalmente una politica che non può uscire dalla regione KNOWN, eliminando per costruzione la dipendenza dal coefficiente di concentrabilità. Il prezzo da pagare è la necessità di selezionare la soglia $\varkappa$ con cura: una soglia troppo bassa restringe eccessivamente la regione esplorabile; una troppo alta lascia passare coppie con incertezza elevata.

5.3 Analisi della scelta della soglia e legame con la teoria dei test statistici

La calibrazione di $\varkappa$ può essere vista come un problema di test statistico. Si definisce l'ipotesi nulla H_0: $(s,a) \in \text{supporto}(d^\beta)$ contro l'alternativa H_1: $(s,a) \notin \text{supporto}(d^\beta)$. Il discriminatore KNOWN basato sull'ensemble implementa un test statistico con livello di significatività α. La scelta di $\varkappa$ corrisponde alla scelta del livello α: $\varkappa$ piccolo (test severo) equivale a α piccolo e bassa potenza statistica contro i falsi positivi (coppie dichiarate note ma in realtà fuori distribuzione). La teoria della stima della densità in dimensione alta (curse of dimensionality) implica che per spazi di stato con dimensione d alta, il test diventa meno affidabile a parità di N. MOReL estende la sua analisi al caso di spazi continui usando estimatori della densità kernel, ma i bound si degradano come $O(N^{-2/(d+4)})$ — convergenza sensibilmente più lenta di quella parametrica $O(1/\sqrt{N})$ valida per spazi finiti.

6. MOBILE: incertezza implicita tramite stima della verosimiglianza e penalità informazionale

MOBILE (Sun et al., 2023 — acronimo di Model-Based Offline Imitation Learning with Implicit Likelihood Estimation) affronta il problema della stima dell'incertezza da una prospettiva radicalmente diversa: invece di usare la varianza dell'ensemble come proxy dell'incertezza epistemica, stima direttamente la log-verosimiglianza negativa delle transizioni osservate sotto il modello appreso, interpretandola come misura di anomalia.

6.1 Il principio della verosimiglianza implicita

Il razionale di MOBILE parte dall'osservazione che la varianza dell'ensemble, pur computazionalmente conveniente, è una misura indiretta e approssimata dell'incertezza del modello. In particolare, non distingue l'incertezza epistemica (dovuta alla mancanza di dati) dall'incertezza aletoristica (dovuta alla stocasticità

intrinseca dell'ambiente). MOBILE propone di usare come misura di incertezza la quantità:

$$u_MOBILE(s,a) = -\mathbb{E}_{s' \sim D(\cdot|s,a)} [\log \hat{P}_\theta(s'|s,a)]$$

che è la cross-entropia empirica tra la distribuzione reale delle transizioni (stimata empiricamente dai dati vicini a (s,a) nel dataset) e la distribuzione predetta dal modello $\hat{P}_\theta$. Quando (s,a) è ben coperto da D, u_MOBILE è piccola perché il modello ha imparato bene la distribuzione locale; quando (s,a) è fuori distribuzione, u_MOBILE è grande perché il modello non ha visto dati sufficienti per stimare accuratamente $P(\cdot|s,a)$.

Per rendere questa misura computazionalmente trattabile, MOBILE adotta un approccio di stima tramite nearest-neighbor: per ogni coppia (s,a), si identificano i k vicini più prossimi nel dataset D secondo una metrica nello spazio stato-azione, e si stima la distribuzione locale delle transizioni tramite la distribuzione empirica su questi k campioni.

6.2 Derivazione della penalità MOBILE e connessione con MOPO

La funzione di ricompensa modificata di MOBILE è:

$$\hat{r}_MOBILE(s,a) = r(s,a) - \lambda \cdot u_MOBILE(s,a)$$

La connessione formale con MOPO emerge dal seguente risultato. Sotto l'ipotesi che il modello $\hat{P}_\theta$ appartenga a una famiglia parametrica gaussiana $\hat{P}_\theta(s'|s,a) = \mathcal{N}(s'; \mu_\theta(s,a), \Sigma_\theta(s,a))$, la misura u_MOBILE si espande come:

$$u_MOBILE(s,a) = \tfrac{1}{2} \operatorname{tr}(\Sigma_\theta^{-1}(s,a) \cdot \mathrm{Cov}_{\mathrm{empirica}}(s,a)) + \tfrac{1}{2} \|\mu_\theta(s,a) - \hat{\mu}_{\mathrm{empirica}}(s,a)\|^2_{\Sigma_\theta^{-1}} + \tfrac{1}{2} \log \det \Sigma_\theta(s,a) + \text{costante}$$

Il primo termine misura il disallineamento tra la varianza predetta dal modello e quella empirica — un indicatore di miscalibrazione; il secondo misura l'errore sistematico nella previsione della media; il terzo penalizza la varianza stessa del modello. Questa decomposizione mostra che u_MOBILE è una forma di distanza di Mahalanobis generalizzata tra la distribuzione predetta e quella empirica, e che cattura sia l'incertezza epistemica sia quella aletoristica in modo più finemente calibrato rispetto alla varianza dell'ensemble.

La garanzia teorica principale di MOBILE stabilisce che, con alta probabilità, u_MOBILE è un limite superiore all'errore di predizione del modello in norma L1 moltiplicato per un fattore dipendente dalla dimensione dello spazio latente e dalla covarianza. Sostituendo questa stima nella derivazione del bound di subottimalità di MOPO si ottiene un bound analogo ma con costanti migliorate.

6.3 Aspetti implementativi e stima della densità nel latente JEPA

Un contributo di rilievo di MOBILE riguarda la sua applicabilità a spazi di stato ad alta dimensione — in particolare le rappresentazioni latenti ottenute da encoder profondi come la JEPA. In tale contesto, l'albero KD (KD-tree) standard per la ricerca dei vicini più prossimi diventa proibitivamente costoso per dimensioni d > 50. MOBILE adotta la ricerca approssimata con indici FAISS (Facebook AI Similarity Search) basati su quantizzazione del prodotto (PQ), che riduce il costo di ricerca da O(N · d) a O(N^{0.5} · d) con perdita controllabile di precisione.

Questo rende MOBILE l'approccio più naturale per l'offline MBRL su rappresentazioni latenti di JEPA: invece di operare nello spazio di osservazione ad alta dimensione, si apprende il modello di transizione direttamente nello spazio latente dell'encoder contestuale (context encoder) di I-JEPA o V-JEPA, si stima u_MOBILE nello spazio latente (dimensione tipicamente tra 256 e 1024), e si penalizza la ricompensa sintetica di conseguenza. Il vantaggio concettuale è che lo spazio latente di JEPA, per costruzione, codifica soltanto le caratteristiche predittivamente rilevanti dell'osservazione — esattamente le caratteristiche che il modello di transizione deve apprendere — eliminando il rumore sensoriale che degrada la qualità delle stime di incertezza nell'osservazione grezza.

7. Teoria della generalizzazione conservativa: unificazione e limiti

I tre algoritmi descritti nelle sezioni precedenti possono essere unificati all'interno di un framework teorico che va sotto il nome di *conservative offline model-based RL*. Il principio unificante è il seguente: si cerca una politica π che massimizza il rendimento pesante su un insieme di MDP plausibili $M_\delta \subseteq M$ anziché sul singolo MDP reale M, dove M_δ è l'insieme di MDP coerenti con i dati osservati a livello di confidenza 1-δ.

Formalmente, la politica conservativa ottimale è:

$$\pi_{conservative} = \arg\max_{\pi} \min_{M' \in M_\delta} J_{M'}(\pi)$$

Questo problema minmax è noto in teoria dei giochi come il problema del *worst-case performance* ed è strettamente connesso alla robustezza distributiva (*distributionally robust optimization*, DRO). Levine et al. (2020) e Rashidinejad et al. (2021) hanno mostrato che la soluzione del problema minmax, sotto ipotesi di convessità dell'insieme M_δ, corrisponde esattamente alla politica ottenuta ottimizzando su un MDP con ricompensa penalizzata dall'incertezza, come in MOPO e MOReL.

Il bound di generalizzazione conservativa più stretto disponibile (Rashidinejad et al., 2021, per MDP tabellari; esteso al caso continuo da Zhan et al., 2022) afferma che la subottimalità di $\pi_{conservative}$ soddisfa:

$$J(\pi^*) - J(\pi_{conservative}) = O\left(\sqrt{C^{\pi^*} \cdot H^3 / N}\right)$$

dove $H = 1/(1-\gamma)$ è l'orizzonte effettivo (finito per $\gamma < 1$) e C^{π^*} è il coefficiente di concentrabilità della politica ottimale rispetto al dataset. Notare che il bound dipende

da C^{π*} — la concentrabilità della politica *ottimale* — e non da quella della politica appresa, il che è un miglioramento sostanziale rispetto ai bound precedenti. Questo risultato è minimax-ottimale nel senso che esiste un'istanza di MDP per la quale qualsiasi algoritmo offline deve pagare almeno $O(\sqrt{(C^{\pi^*} \cdot H^3 / N)})$ di subottimalità.

7.1 La dicotomia copertura completa vs. copertura parziale

La teoria della generalizzazione conservativa distingue due regimi fondamentali. Nel regime di *copertura completa* (*full coverage*), il dataset soddisfa $d^\beta(s,a) \geq c > 0$ per ogni (s,a) rilevante per π*, ovvero il dataset comportamentale copre l'intera regione di interesse. In questo regime C^{π*} è limitato da 1/c e il bound di subottimalità decresce come $O(1/\sqrt{N})$, garantendo una convergenza classica.

Nel regime di *copertura parziale* (*partial coverage*), il dataset non copre alcune coppie rilevanti per π*, e C^{π*} può crescere senza limite. In questo regime il bound di subottimalità può diventare arbitrariamente grande per qualsiasi algoritmo — inclusi MOPO, MOReL e MOBILE. Questo risultato negativo è fondamentale: esso dimostra che l'offline MBRL non può, in linea di principio, garantire la qualità della politica appresa senza ipotesi sulla copertura del dataset. La copertura parziale è la regola, non l'eccezione, nelle applicazioni reali.

7.2 Ipotesi di regolarità e sfruttamento della struttura del modello

Un modo per ovviare ai limiti della copertura parziale consiste nell'aggiungere ipotesi di regolarità sulla struttura del MDP o del modello del mondo che consentano la generalizzazione attraverso lo spazio degli stati. L'idea è che, se la funzione di transizione P è Lipschitz con costante L_P rispetto a (s,a), allora una coppia fuori distribuzione $(\tilde{s}, \tilde{a})$ con distanza ε da una coppia osservata (s, a) ha una transizione che differisce al più di L_P · ε in norma L1. Questo permette di trasferire la conoscenza acquisita sulla coppia osservata alla coppia non osservata con un errore controllato dalla Lipschitzianità del modello.

Formalizzando questo argomento, si ottiene un bound migliorato nel quale il coefficiente di concentrabilità C^{π*} viene rimpiazzato da:

$$\tilde{C}^{\pi^*} = C^{\pi^*} / (1 + L_P \cdot r_{cover})$$

dove r_cover è il raggio della copertura del dataset (il raggio massimo delle palle intorno ai punti del dataset che contengono almeno un punto per ogni (s,a) rilevante per π*). Quando il dataset è denso (r_cover piccolo) e il modello è Lipschitz (L_P grande ma finito), $\tilde{C}^{\pi^*} \ll C^{\pi^*}$.

Nel contesto delle rappresentazioni latenti di JEPA, questa ipotesi di Lipschitzianità è particolarmente naturale: la struttura del training JEPA — che predice nella norma dello spazio latente — induce implicitamente una forma di regolarità sulla topologia dello spazio delle rappresentazioni. Se lo spazio latente di JEPA è una varietà riemanniana con curvatura sezionale limitata (come suggerito dall'analisi geometrica

del Capitolo IX di questa monografia), allora le transizioni latenti sono automaticamente Lipschitz con costante che dipende dalla curvatura e dal raggio di ingetttività della varietà.

8. Offline MBRL con rappresentazioni latenti JEPA: sinergie e direzioni aperte

La convergenza tra il paradigma dell'offline MBRL e le architetture JEPA per l'apprendimento delle rappresentazioni è una delle direzioni di ricerca più promettenti del campo. In questo contesto si possono distinguere due modalità di integrazione.

La prima modalità è *offline MBRL con encoder fisso*: si addestra prima l'encoder di JEPA (I-JEPA o V-JEPA) su un dataset non etichettato (eventualmente distinto dal dataset offline di RL), poi si apprende un modello di transizione lineare o non-lineare nello spazio latente, e infine si applica MOPO o MOReL nel latente. Il vantaggio è la separazione netta tra apprendimento della rappresentazione e apprendimento del comportamento; lo svantaggio è che l'encoder ottimizzato per la previsione predittiva potrebbe non codificare tutte le variabili di stato rilevanti per il controllo.

La seconda modalità è *offline MBRL con encoder adattivo*: si co-addestrano encoder, modello di transizione latente e politica end-to-end su D, con la penalità di MOBILE calcolata direttamente nel latente. Questo approccio massimizza la pertinenza delle rappresentazioni per il controllo ma introduce instabilità di training dovuta alla non-stazionarietà della distribuzione dell'encoder durante l'apprendimento.

Un problema formale aperto riguarda la definizione della misura di incertezza u nel latente quando l'encoder è esso stesso incerto. In MOPO e MOReL, l'incertezza del modello di transizione è l'unica fonte di penalizzazione; ma quando l'encoder φ è parametrico e appreso, anche la rappresentazione $\hat{z} = \varphi(o)$ è incerta. La propagazione di questa incertezza attraverso l'encoder richiede strumenti della teoria della propagazione degli errori nelle reti neurali — in particolare, la Jacobian-based uncertainty propagation — che rimane un problema computazionalmente costoso e teoricamente non risolto nel caso delle reti neurali profonde non-lineari.

9. Valutazione empirica e i benchmark D4RL

La valutazione sistematica degli algoritmi di offline MBRL si svolge prevalentemente sulle suite di benchmark D4RL (Fu et al., 2020), che raccoglie dataset di qualità eterogenea per vari ambienti di locomozione e manipolazione. I dataset di D4RL si distinguono in base alla qualità della politica comportamentale β: *random* (azioni casuali), *medium* (politica mediocre addestrata online), *medium-replay* (tutti i dati collezionati durante l'addestramento di una politica medium), *expert* (politica ottimale), e *medium-expert* (misto).

I risultati sperimentali su D4RL evidenziano le seguenti regolarità. MOPO supera i metodi offline model-free di riferimento (BCQ, CQL, IQL) nei dataset medium e

medium-replay, dove la politica comportamentale è informativa ma non ottimale; i guadagni sono particolarmente evidenti su HalfCheetah e Walker2d, ambienti con dinamica sufficientemente liscia da consentire rollout sintetici accurati. MOReL ottiene risultati comparabili ma con maggiore varianza inter-seed, dovuta alla sensibilità alla soglia κ. MOBILE ottiene il miglior punteggio assoluto nella maggior parte delle configurazioni di AntMaze, che è l'ambiente con la maggiore sparsità della ricompensa e la più alta dimensionalità dello spazio degli stati tra quelli inclusi in D4RL.

Va tuttavia osservato — e questo è un punto metodologico che la comunità ha tardato a riconoscere — che i punteggi di D4RL sono normalizzati rispetto alla performance di una politica casuale e di una politica esperta addestrata online: il punteggio 100 corrisponde alla performance dell'esperto. I metodi offline MBRL raramente superano 85-90 anche nelle configurazioni più favorevoli, il che significa che il gap rispetto alla performance online rimane sostanziale. Questo gap riflette esattamente i limiti teorici della copertura parziale discussi nella Sezione 7.1 e non è colmabile senza interazione con l'ambiente o senza assunzioni strutturali aggiuntive.

10. Connessione con la valutazione sistematica dei world model come problema statistico

L'offline MBRL pone in evidenza un problema generale che la monografia tratta sparsamente in altri capitoli ma che merita una considerazione esplicita: la valutazione sistematica dei modelli del mondo. Nell'offline MBRL, la qualità del modello di transizione non è separabile dalla qualità della politica prodotta — un modello più accurato non produce necessariamente una politica migliore, perché l'accuratezza viene misurata sulla distribuzione del dataset, non sulla distribuzione indotta dalla politica ottimizzata.

Questo crea un paradosso valutativo: le metriche di accuratezza del modello standard (MSE sulla transizione, log-verosimiglianza del dataset, ELBO variazionale) non predicono affidabilmente la performance in controllo. Il fenomeno è stato documentato empiricamente da Matsushima et al. (2021) e formalizzato da Lambert et al. (2022) attraverso la nozione di *model alignment*: un modello del mondo è allineato se e solo se la distribuzione degli stati sotto la politica ottimale nel modello coincide con la distribuzione degli stati sotto la politica ottimale nel mondo reale. La misura di misalignment è il coefficiente di concentrabilità che compare nei bound di subottimalità — il che chiude il cerchio teorico.

La valutazione sistematica dei world model come disciplina autonoma richiederebbe dunque metriche che catturino direttamente il misalignment tra distribuzione del modello e distribuzione del mondo reale sotto la politica ottimale — una quantità che non è misurabile senza interazione con l'ambiente reale, e che quindi rimane strutturalmente inaccessibile nel setting puramente offline. Questo limite fondamentale giustifica l'interesse crescente per i paradigmi di offline-to-online e offline-finetuning, nei quali la politica appresa offline viene affinata con un numero

limitato di interazioni reali, sfruttando la politica offline come punto di partenza per ridurre il costo esplorazione.

XXII. Sicurezza certificata e robustezza formale dei sistemi autonomi con modelli del mondo

1. Il problema della sicurezza come questione tecnica formale, non come addendum etico

La riflessione sulla sicurezza dei sistemi autonomi che apprendono e pianificano mediante modelli del mondo è stata storicamente relegata ai margini della trattazione tecnica — come appendice etica, come sezione delle *broader impacts*, come nota a piè di pagina dedicata alle implicazioni sociali del lavoro. Questa collocazione è epistemicamente errata e praticamente pericolosa. La sicurezza in deployment di un sistema che pianifica su un modello del mondo appreso è un problema tecnico formale con una propria letteratura, un proprio apparato matematico e i propri teoremi di impossibilità, del tutto analoga — nella struttura — ai problemi di ottimizzazione, inferenza e rappresentazione trattati nei capitoli precedenti.

La tesi centrale di questo capitolo è che la sicurezza certificata dei sistemi autonomi con modelli del mondo richiede la risposta a tre domande formalmente distinte. La prima: il sistema produrrà un'azione sicura con alta probabilità, anche quando le sue rappresentazioni interne sono approssimate? Questa è la domanda della *copertura statistica* e viene affrontata dalla predizione conforme (*conformal prediction*). La seconda: esiste un insieme di stati nel quale il sistema è garantito a non entrare mai, indipendentemente dalle azioni dell'ambiente? Questa è la domanda della *raggiungibilità* (*reachability*) e viene affrontata dall'analisi di raggiungibilità Hamilton-Jacobi e dalle funzioni barriera di controllo (*control barrier functions*, CBF). La terza: la politica appresa soddisfa formalmente una specifica comportamentale espressa in logica temporale? Questa è la domanda della *verifica formale* (*formal verification*) e viene affrontata dalla sintesi di shield e dalla certificazione di Lyapunov delle politiche neurali.

Queste tre domande non sono tra loro indipendenti: le garanzie statistiche della predizione conforme forniscono gli insiemi di fiducia sugli stati predetti che alimentano l'analisi di raggiungibilità; le funzioni barriera di controllo possono essere incorporate come vincoli nella pianificazione latente del modello del mondo; la verifica formale può essere applicata alla composizione modello-politica nel latente anziché all'agente nel mondo reale. Il capitolo sviluppa questa struttura concettuale con il rigore matematico che la materia richiede.

2. La predizione conforme applicata ai sistemi di controllo: garanzie di copertura distribuzione-libera

La predizione conforme (*conformal prediction*, CP) è un framework statistico che consente di costruire insiemi di predizione con garanzia di copertura marginale valida senza alcuna ipotesi distributiva sul processo generativo dei dati. Introdotta da Vovk, Gammerman e Shafer nel 2005 nella sua formulazione più generale, la CP è emersa come strumento fondamentale per la calibrazione dell'incertezza dei sistemi di apprendimento automatico in contesti safety-critical.

2.1 Predizione conforme in split: formulazione base

Sia $(X_1, Y_1), ..., (X_n, Y_n), (X_{n+1}, Y_{n+1})$ una sequenza di coppie input-output scambievoli (exchangeable), dove $X \in \mathcal{X}$ e $Y \in \mathcal{Y}$. Un predittore base $\hat{g}: \mathcal{X} \to \mathcal{Y}$ è disponibile (tipicamente una rete neurale o un modello del mondo), e si desidera costruire una regione di predizione $\hat{C}(X_{n+1}) \subseteq \mathcal{Y}$ tale che:

$$\mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) \geq 1 - \alpha$$

per un livello di rischio $\alpha \in (0,1)$ prefissato, senza ipotesi sulla distribuzione di (X, Y).

La CP in split (split conformal prediction) divide i dati in un insieme di calibrazione $\mathcal{D}_{cal} = \{(X_1, Y_1), ..., (X_n, Y_n)\}$ e un punto di test (X_{n+1}, Y_{n+1}). Si calcola uno *score di non-conformità* $s: \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ tale che $s(X, Y)$ sia piccolo quando Y è una risposta tipica per l'input X secondo il predittore $\hat{g}$ (ad esempio, $s(X,Y) = ||Y - \hat{g}(X)||_2$ per predizioni continue). Si calcolano gli score sull'insieme di calibrazione:

$$s_i = s(X_i, Y_i), i = 1, ..., n$$

e si definisce il quantile empirico al livello $\lceil (1-\alpha)(n+1) \rceil / n$:

$$\hat{q}_{1-\alpha} = \text{Quantile}(\{s_1, ..., s_n\}, \lceil (1-\alpha)(n+1) \rceil / n)$$

L'insieme di predizione conforme è allora:

$$\hat{C}(X_{n+1}) = \{y \in \mathcal{Y} : s(X_{n+1}, y) \leq \hat{q}_{1-\alpha}\}$$

Il teorema fondamentale della CP afferma che, sotto la sola ipotesi di scambiabilità delle coppie:

$$1 - \alpha \leq \mathbb{P}(Y_{n+1} \in \hat{C}(X_{n+1})) \leq 1 - \alpha + 1/(n+1)$$

La garanzia è *marginale* (media sulla distribuzione di tutti i possibili punti di test) ma non richiede alcuna ipotesi di buona specifica del modello $\hat{g}$, di gaussianità, di lipschitzianità, o di qualsiasi altra proprietà distributiva. Questo la rende

fondamentalmente diversa dai bound dell'ensemble bayesiano o dell'ensemble di modelli, che richiedono ipotesi distribuzionali non verificabili in pratica.

2.2 Predizione conforme per sistemi dinamici: la sfida della scambiabilità sequenziale

L'applicazione della CP ai sistemi di controllo con modelli del mondo incontra immediatamente il problema della scambiabilità: le traiettorie di un sistema dinamico sono *sequenze* di coppie (s_t, a_t) correlate temporalmente, non coppie i.i.d. La violazione della scambiabilità invalida la garanzia teorica della CP standard.

Angelopoulos et al. (2022) e Dixit et al. (2023) hanno affrontato questo problema estendendo la CP al contesto sequenziale attraverso due approcci complementari.

Il primo approccio, denominato *CP conforme online*, aggiorna il quantile di calibrazione in modo adattivo a ogni passo temporale. Sia α_t il livello di confidenza al passo t e siano $s_1, ..., s_t$ gli score di non-conformità osservati fino al passo t. Si aggiorna il livello di rischio secondo la regola:

$$\alpha_{t+1} = \alpha_t + \eta \cdot (\alpha - \mathbb{1}[Y_t \notin \hat{C}_t(X_t)])$$

dove $\eta > 0$ è un tasso di apprendimento. Questo meccanismo di retroazione forza la copertura empirica verso il livello target α nel lungo termine, nel senso che:

$$(1/T) \cdot \Sigma_{t=1}^{T} \mathbb{1}[Y_t \notin \hat{C}_t(X_t)] \to \alpha \text{ per } T \to \infty$$

sotto condizioni di stazionarietà debole della sequenza. La garanzia è *time-averaged* anziché marginale, ma è comunque distribuzione-libera e non richiede modelli del processo stocastico sottostante.

Il secondo approccio è la *CP per sistemi dinamici stazionari* di Cleaveland et al. (2023), che sfrutta il *mixing* del processo per ridurre la dipendenza temporale a una quantità controllabile. Sia τ_{mix} il tempo di mixing del processo (il tempo minimo tale che la dipendenza tra s_t e $s_{t+\tau}$ sia trascurabile). Se si usa uno score di non-conformità calcolato su finestre temporali non sovrapposte di lunghezza $L \gg \tau_{mix}$, le finestre diventano approssimativamente scambiabili e la garanzia della CP standard si applica con una correzione dipendente dal mixing.

2.3 CP applicata al modello di transizione del world model

Nel contesto specifico dei world model, la CP si applica naturalmente alla stima dell'incertezza del modello di transizione. Sia $P_\theta: \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ il modello appreso (ad esempio l'RSSM di Dreamer o un modello lineare nello spazio latente di JEPA). Si definisce lo score di non-conformità del modello come:

$$s(s_t, a_t, s_{t+1}) = ||s_{t+1} - \mathbb{E}_{P_\theta}[s' | s_t, a_t]||_2$$

Gli score vengono calcolati su un insieme di calibrazione $\mathcal{D}$_cal di traiettorie reali. Il quantile conforme $\hat{q}_{1-\alpha}$ definisce un insieme di predizione per il successore:

$$\hat{C}_\theta(s_t, a_t) = \{s' \in \mathcal{S} : ||s' - \mathbb{E}_{P_\theta}[s' | s_t, a_t]||_2 \leq \hat{q}_{1-\alpha}\}$$

che è una palla nello spazio degli stati di raggio adattivo $\hat{q}_{1-\alpha}$ e contiene lo stato successore reale con probabilità almeno $1-\alpha$, indipendentemente dalla qualità del fit del modello.

Questo risultato ha un'implicazione fondamentale per la pianificazione sicura: invece di pianificare su un singolo stato futuro previsto, il sistema deve pianificare considerando l'intero insieme $\hat{C}_\theta(s_t, a_t)$ come possibile stato successore. La pianificazione *worst-case* su questo insieme produce politiche che sono *robuste* all'errore del modello con garanzia probabilistica $1-\alpha$ — una proprietà che nessun ensemble bayesiano, nessuna varianza dell'ensemble, nessuna incertezza MC-Dropout può garantire senza ipotesi distribuzionali.

2.4 Mondrian CP e garanzie condizionali

Una limitazione della CP standard è che la garanzia di copertura è *marginale* — vale in media sulla distribuzione degli input, ma può non valere condizionatamente a specifici valori di X. In un sistema di controllo, questo significa che la garanzia di sicurezza potrebbe non valere nelle regioni di spazio degli stati più critiche (ad esempio, vicino a ostacoli o in situazioni di emergenza), che sono precisamente le regioni in cui è più necessaria.

La *CP di Mondrian* (Vovk et al., 2003) affronta questo problema partizionando lo spazio degli input in categorie omogenee e applicando la CP separatamente a ciascuna categoria. Se le categorie sono definite da una partizione $\mathcal{K} = \{K_1, ..., K_m\}$ dello spazio degli stati, e se si calcola un quantile di calibrazione separato $\hat{q}_{1-\alpha}^{k}$ per ogni categoria k, allora la garanzia condizionale vale:

$$\mathbb{P}(Y_{n+1} \in \hat{C}^{\kappa(X_{n+1})}(X_{n+1}) \mid X_{n+1} \in K_{\kappa(X_{n+1})}) \geq 1 - \alpha$$

per ogni categoria k tale che l'insieme di calibrazione abbia almeno un campione in K_k. La granularità della partizione $\mathcal{K}$ controlla il bias-varianza della garanzia: partizioni fini producono garanzie più precise ma richiedono più dati di calibrazione per categoria.

Nel contesto del controllo robotico con world model, una partizione naturale è quella indotta dalle regioni dello spazio degli stati con proprietà dinamiche omogenee — ad esempio, le regioni di stabilità del sistema di controllo, le vicinanze agli ostacoli, o le regioni di bassa copertura del dataset offline discusse nel capitolo precedente. La CP di Mondrian fornisce quindi una ponte teorico tra l'incertezza del modello del mondo e le garanzie di sicurezza condizionate al contesto operativo.

3. Analisi di raggiungibilità Hamilton-Jacobi: insiemi sicuri e loro calcolo

L'analisi di raggiungibilità studia insiemi di stati raggiungibili e non raggiungibili sotto la dinamica di un sistema, con o senza il controllo di un agente. Nel contesto della sicurezza dei sistemi autonomi, l'obiettivo è caratterizzare l'*insieme sicuro* — la regione dello spazio degli stati dalla quale il sistema può garantire di non violare mai una specifica di sicurezza (vincolo sullo stato) indipendentemente dall'azione degli agenti perturbatori.

3.1 Il problema di raggiungibilità backward e la formulazione Hamilton-Jacobi

Si consideri un sistema con dinamica continua:

$\dot{x}(t) = f(x(t), u(t), d(t))$

dove $x \in \mathbb{R}^n$ è lo stato, $u \in \mathcal{U}$ è il controllo del sistema, e $d \in \mathcal{D}$ è un ingresso di perturbazione (o l'azione di un avversario). L'insieme di sicurezza è descritto come il sopralivello di una funzione vincolo:

$\mathcal{C} = \{x \in \mathbb{R}^n : g(x) \leq 0\}$

Il problema di raggiungibilità backward chiede: dato l'orizzonte temporale $T > 0$ e l'insieme del target $\mathcal{T}$ (che rappresenta le violazioni di sicurezza, tipicamente il complemento di $\mathcal{C}$), qual è l'insieme di tutti gli stati da cui è impossibile evitare di raggiungere $\mathcal{T}$ entro il tempo T, anche con il controllo ottimale del sistema?

Questo insieme — chiamato *backward reachable set* o *avoid set* — è caratterizzato dalla funzione valore del seguente problema di controllo ottimale in orizzonte finito:

$V(x, t) = \min_u \max_d \min_{\{\tau \in [t, T]\}} g(x(\tau))$

dove il min esterno è sull'ottimizzazione del controllo (il sistema vuole evitare la violazione), il max è sulla perturbazione avversariale (l'ambiente vuole indurre la violazione), e il min interno è sul minimo valore del vincolo lungo la traiettoria (la violazione avviene se il minimo è positivo).

La funzione V è la soluzione viscosa dell'equazione di Hamilton-Jacobi-Isaacs (HJI):

$\partial V/\partial t + H(x, t, \nabla_x V) = 0 \text{ su } [0,T) \times \mathbb{R}^n$

$V(x, T) = g(x)$

dove l'Hamiltoniana del gioco differenziale è:

$H(x, t, p) = \min_u \max_d [p^T f(x, u, d)]$

con $p = \nabla_x V$ il gradiente della funzione valore (il costato nel linguaggio del controllo ottimale). Il *backward reachable set* all'istante $t = 0$ è:

$$\mathcal{R}(\mathcal{T}, T) = \{x \in \mathbb{R}^n : V(x, 0) > 0\}$$

Tutti gli stati in $\mathcal{R}(\mathcal{T}, T)$ sono *non sicuri*: da essi, anche il controllo ottimale non può evitare la violazione entro T. Il complemento $S(T) = \mathbb{R}^n \setminus \mathcal{R}(\mathcal{T}, T)$ è l'insieme sicuro.

3.2 La soluzione computazionale e le sue limitazioni in alta dimensione

La soluzione della PDE di HJI è computazionalmente intrattabile per dimensioni $n > 6$ con i metodi classici di discretizzazione della griglia (Bansal et al., 2017). I metodi basati su griglia (toolbox Level Set di Mitchell, 2005) scalano esponenzialmente con n, rendendo inapplicabile l'approccio diretto ai sistemi robotici con stati ad alta dimensione.

Tre direzioni di ricerca affrontano questo problema. La prima è la *decomposizione per sistemi accoppiati debolmente*: se la dinamica è scomponibile in sottosistemi di bassa dimensione con accoppiamento limitato, il calcolo del reachable set si fattorizza approssimativamente sul prodotto dei sottosistemi. Sia $f(x, u) = (f_1(x_1, u), f_2(x_2, u) + g(x_1))$ con g piccolo in norma: si calcola V_1 e V_2 separatamente e si usa un argomento di perturbazione per stimare l'errore indotto dall'accoppiamento.

La seconda direzione è l'*approssimazione neurale dell'HJI* (Bansal e Tomlin, 2021; Everett et al., 2021): si parametrizza la funzione valore come una rete neurale V_θ e si addestra minimizzando il residuo della PDE:

$$\mathcal{L}_{HJI}(\theta) = \mathbb{E}_{(x,t) \sim \mathcal{D}_{colloc}} [(\partial V_\theta/\partial t + H(x, t, \nabla_x V_\theta))^2] + \mathbb{E}_{x \sim \mathbb{R}^n} [(V_\theta(x, T) - g(x))^2]$$

dove $\mathcal{D}_{colloc}$ è una distribuzione di punti di collocazione nello spazio-tempo. Questo approccio è ispirato ai Physics-Informed Neural Networks (PINN) di Raissi et al. (2019) e scala polinomialmente con n, ma perde la garanzia di soluzione viscosa: la rete neurale è un'approssimazione che può avere errori non uniformi nello spazio.

La terza direzione, più conservativa ma formalmente garantita, è la *reachability per funzioni di Lyapunov e CBF*.

3.3 Funzioni barriera di controllo: dalla reachability alle garanzie in tempo reale

Le *funzioni barriera di controllo* (Control Barrier Functions, CBF) furono introdotte da Ames et al. (2017) come strumento per garantire la sicurezza in tempo reale senza risolvere l'intera PDE di HJI. Una CBF è una funzione $h: \mathbb{R}^n \to \mathbb{R}$ tale che:

(i) L'insieme sicuro $\mathcal{C} = \{x : h(x) \geq 0\}$ è compatto e non vuoto (ii) Per ogni $x \in \partial\mathcal{C}$ (il bordo dell'insieme sicuro), esiste $u \in \mathcal{U}$ tale che $\dot{h}(x, u) \geq -\alpha(h(x))$

dove $\dot{h}(x, u) = \partial h/\partial x \cdot f(x, u)$ è la derivata di Lie di h lungo le traiettorie del sistema, e $\alpha: \mathbb{R} \to \mathbb{R}$ è una funzione di classe K estesa (strettamente crescente, $\alpha(0) = 0$).

Il teorema fondamentale delle CBF afferma che se h è una CBF per il sistema e se per ogni $t \geq 0$ si applica un'azione u_t che soddisfa la *condizione CBF*:

$\dot{h}(x(t), u_t) + \alpha(h(x(t))) \geq 0$

allora $x(0) \in \mathscr{C} \Rightarrow x(t) \in \mathscr{C}$ per ogni $t \geq 0$. In altri termini, $\mathscr{C}$ è *positivamente invariante* sotto qualsiasi controllo che soddisfi la condizione CBF.

In pratica, si formula un *filtro CBF*: data un'azione nominale u_nom (prodotta dalla politica appresa del modello del mondo), si risolve in tempo reale il QP (Quadratic Program):

$u^* = \text{argmin}_{u \in \mathscr{U}} ||u - u_{nom}||^2 \text{ s.t. } \dot{h}(x, u) + \alpha(h(x)) \geq 0$

Il QP ha una soluzione analitica nella maggior parte dei casi e può essere risolto in millisecondi su hardware embedded. Il risultato è una modifica minimale dell'azione nominale che garantisce la sicurezza senza richiedere la re-ottimizzazione della politica.

3.4 CBF adattive e incertezza del modello del mondo

La teoria classica delle CBF ipotizza che il modello della dinamica f sia noto esattamente. Quando f è appreso da dati — come nel caso dei modelli del mondo neurali — si introduce un errore di modellazione che può invalidare la garanzia di invarianza positiva.

Sia $\hat{f}(x, u)$ il modello appreso (ad esempio, il modello di transizione lineare nello spazio latente di JEPA) e sia $\Delta f(x, u) = f(x, u) - \hat{f}(x, u)$ l'errore di modello. La condizione CBF perturbata diventa:

$\dot{h}(x, u) = \partial h/\partial x \cdot f(x, u) = \partial h/\partial x \cdot \hat{f}(x, u) + \partial h/\partial x \cdot \Delta f(x, u) \geq \partial h/\partial x \cdot \hat{f}(x, u) - ||\partial h/\partial x||_2 \cdot ||\Delta f(x, u)||_2$

Per garantire $\dot{h} + \alpha(h) \geq 0$ nonostante l'incertezza, si modifica la condizione CBF come:

$\partial h/\partial x \cdot \hat{f}(x, u) + \alpha(h(x)) - ||\partial h/\partial x||_2 \cdot B(x, u) \geq 0$

dove $B(x, u)$ è un limite superiore a $||\Delta f(x, u)||_2$. Questo limite superiore è esattamente l'informazione prodotta dalla CP applicata al modello di transizione: $B(x, u) = \hat{q}_{1-\alpha}^{\kappa(x)}$ (il quantile conforme nella categoria Mondrian a cui appartiene x). La *CBF adattiva conforme* (Dixit et al., 2023) combina quindi la CP e le CBF in un framework unificato che garantisce la sicurezza con probabilità $1-\alpha$ senza ipotesi distribuzionali sul processo stocastico.

Il QP modificato diventa:

$u^* = \text{argmin}_{u \in \mathcal{U}} ||u - u_{nom}||^2 \text{ s.t. } \partial h/\partial x \cdot f(x, u) + \alpha(h(x)) \geq ||\partial h/\partial x||_2 \cdot \hat{q}_{1-\alpha}^{\kappa(x)}$

La garanzia risultante è: se $x(0) \in \mathcal{C}$ e si applica u^* a ogni passo, allora:

$\mathbb{P}(x(t) \in \mathcal{C} \text{ per ogni } t \in [0, T]) \geq (1-\alpha)^{T/\Delta t}$

dove Δt è il passo temporale di controllo. Per $\alpha = 0.01$ e $T/\Delta t = 100$ passi, la probabilità di sicurezza sulla traiettoria intera è $(1-0.01)^{100} \approx 0.366$, il che mostra la degradazione esponenziale della garanzia con l'orizzonte. Questo è un limite intrinseco dell'approccio step-by-step e motiva la ricerca di garantie sull'intera traiettoria via la reachability HJI piuttosto che via le CBF.

4. Verifica formale delle politiche neurali: Lyapunov, shield synthesis e LTL

La verifica formale — la dimostrazione matematica che un sistema soddisfa una specifica comportamentale — rappresenta il livello più alto di garanzia di sicurezza. Nel contesto dei sistemi di controllo classici, la verifica formale si applica a politiche con struttura algebraica nota (controllori lineari, automi a stati finiti, sistemi ibridi). L'applicazione ai sistemi con politiche rappresentate da reti neurali profonde — che sono funzioni continue non-lineari di alta dimensione senza struttura algebrica semplice — ha richiesto lo sviluppo di nuovi strumenti.

4.1 Certificazione di stabilità tramite funzioni di Lyapunov neurali

La stabilità di Lyapunov è la forma più classica di verifica formale per i sistemi dinamici. Una funzione $V: \mathbb{R}^n \to \mathbb{R}_{\geq 0}$ è una *funzione di Lyapunov* per il sistema $\dot{x} = f(x, \pi(x))$ (dove π è la politica) se:

(i) $V(0) = 0$ e $V(x) > 0$ per $x \neq 0$ (definitezza positiva) (ii) $\nabla V(x) \cdot f(x, \pi(x)) < 0$ per $x \neq 0$ (decrescita stretta lungo le traiettorie)

L'esistenza di V certifica la stabilità asintotica globale del sistema in anello chiuso con la politica π. La sfida computazionale è trovare V tale che la condizione (ii) sia verificabile su tutto $\mathbb{R}^n$.

Chang et al. (2019) e Dawson et al. (2023) hanno proposto di parametrizzare V come una rete neurale V_φ e di ricercare φ tramite un algoritmo iterativo che alterna tra:

(a) Addestramento di V_φ per soddisfare le condizioni (i) e (ii) su un insieme di punti di addestramento $\mathcal{D}_{train}$ (b) Ricerca di *controesempi* — punti x per cui le condizioni vengono violate — tramite un ottimizzatore basato su discesa del gradiente o un SMT-solver (Satisfiability Modulo Theories)

Quando il passo (b) non trova controesempi, si ha una *quasi-certificazione* della stabilità: la politica π è stabile su tutti i punti di $\mathcal{D}_{train}$, ma non necessariamente su tutto lo

spazio. La certificazione completa richiede strumenti di verifica formale per reti neurali, descritti nella sezione seguente.

4.2 Verifica formale di reti neurali con ReLU: MILP e propagazione degli intervalli

La verifica formale di una rete neurale a ReLU — provare che essa soddisfa una proprietà della forma "se $x \in \mathcal{X}$, allora $\pi(x) \in \mathcal{Y}$" — si riduce a un problema di programmazione mista intera-lineare (MILP). Per ogni neurone con attivazione ReLU, si introduce una variabile binaria $\delta \in \{0,1\}$ che indica se la ReLU è attiva ($\delta = 1$, neurone in regime lineare) o inattiva ($\delta = 0$, neurone saturo a zero). Il preattivazione z e l'attivazione a = ReLU(z) soddisfano le coppie di disuguaglianze lineari:

a ≥ 0 a ≥ z a ≤ z - z^{lb}(1 - δ) a ≤ z^{ub} · δ

dove z^{lb} e z^{ub} sono limiti inferiori e superiori sul preattivazione, che vengono calcolati ricorsivamente per propagazione degli intervalli (*interval arithmetic propagation*).

Per una rete con L strati, n neuroni per strato e N input, il MILP ha O(L · n) variabili intere, rendendolo NP-difficile nel caso peggiore. Nella pratica, i solver MILP moderni (Gurobi, CPLEX) con le tecniche di *branch-and-bound* riescono a verificare reti con centinaia di neuroni per strato in tempi ragionevoli per lo spazio dei problemi di verifica rilevanti nel controllo. L'approccio è stato standardizzato nei tools ERAN (Singh et al., 2018), α-β-CROWN (Xu et al., 2021) e Marabou (Katz et al., 2019).

La scalabilità ai sistemi con world model pone tuttavia sfide aggiuntive: la politica da verificare può avere milioni di parametri, e lo spazio degli stati può essere continuo e ad alta dimensione. Le tecniche di *abstract interpretation* (interpretazione astratta) e *bound propagation* affrontano questo problema approssimando i set di raggiungibilità con geometrie convesse (poliedri, zonotopi, ellissoidi) che si propagano analiticamente attraverso gli strati della rete con complessità polinomiale nella dimensione dello spazio latente.

4.3 Shield synthesis e la logica temporale lineare

La *sintesi di shield* (*shield synthesis*) è un approccio alla verifica in tempo reale ispirato ai controller correttori dei sistemi di sicurezza industriale. Un *shield* è un modulo che osserva il comportamento della politica del modello del mondo in tempo reale e corregge le azioni che violerebbero una specifica di sicurezza, nel modo meno invasivo possibile.

La specifica di sicurezza è espressa in *logica temporale lineare* (LTL), una logica proposizionale estesa con modalità temporali. I costrutti fondamentali di LTL sono:

Gφ — "sempre φ" (globally): φ deve valere in ogni istante futuro **F**φ — "eventualmente φ" (finally): φ deve valere in almeno un istante futuro **X**φ — "al prossimo passo φ" (next): φ deve valere all'istante successivo φ **U** ψ — "φ finché non vale ψ" (until): φ vale continuamente finché ψ non diventa vera

Una specifica di sicurezza tipica per un robot in navigazione è:

$$\mathbf{G}\,\neg\,\text{collision} \;\wedge\; \mathbf{G}(\text{goal_requested} \rightarrow \mathbf{F}\,\text{goal_reached})$$

che afferma che il robot non deve mai collidere con ostacoli e che ogni richiesta di raggiungere un obiettivo deve essere eventualmente soddisfatta.

La *sintesi di uno shield dalla specifica LTL* si svolge in due fasi. Nella prima, la formula LTL viene convertita in un automa di Büchi equivalente attraverso la costruzione di Vardi e Wolper (1986). Nella seconda, l'automa viene determinitizzato e utilizzato come controllore sintattico che, a ogni stato del sistema, computa l'insieme di azioni sicure compatibili con la specifica.

L'interfaccia con il sistema di apprendimento avviene tramite il *problema della deviazione minima*: dato un insieme di azioni sicure $\mathcal{U}$_safe(x, q) (dove q è lo stato dell'automa dello shield), si risolve:

$$u^* = \text{argmin}_\{u \in \mathcal{U}\text{_safe}(x,q)\} \; ||u - u_{nom}||_2$$

dove u_nom è l'azione nominale della politica del modello del mondo. Il meccanismo è concettualmente analogo al filtro CBF della Sezione 3.3, ma opera su una specifica formale più espressiva di una singola funzione barriera scalare.

5. Sicurezza nello spazio latente: verifica formale per world model

Le tecniche di verifica formale presentate nelle sezioni precedenti sono tipicamente formulate nello spazio degli stati fisici osservabili. Quando il sistema di controllo pianifica nello spazio latente di un world model — come in Dreamer, MuZero o in un sistema JEPA-based — sorge il problema della *sicurezza latente*: le specifiche di sicurezza sono espresse nel mondo fisico, ma la pianificazione avviene nel latente. La discrepanza tra i due spazi introduce possibilità di violazione che le tecniche standard non possono rilevare.

5.1 Il problema dell'inverso del decodificatore e la sicurezza latente

Si consideri un world model con encoder $\varphi: \mathcal{S} \rightarrow \mathcal{Z}$ e decodificatore $\psi: \mathcal{Z} \rightarrow \mathcal{S}$ (approssimativamente inversi l'uno dell'altro). Una specifica di sicurezza nello spazio fisico $\mathcal{S}$ è dell'insieme $\mathcal{C} = \{s \in \mathcal{S} : g(s) \leq 0\}$. L'insieme sicuro nello spazio latente corrispondente è:

$$\mathcal{C}\text{_latent} = \{z \in \mathcal{Z} : g(\psi(z)) \leq 0\}$$

Poiché ψ è una rete neurale profonda, $\mathcal{C}$_latent ha una geometria complessa — non è in generale un ellissoide, un politopo o un'altra forma convessa semplice. La propagazione della specifica di sicurezza dallo spazio fisico al latente richiede la

preimage computation di $\mathscr{C}$ attraverso ψ, che è esattamente il problema di verifica formale delle reti neurali discusso nella Sezione 4.2.

Il framework di verifica ERAN e α-β-CROWN possono calcolare sovra-approssimazioni conservatrici di $\varphi(\mathscr{C})$ e $\psi^{-1}(\mathscr{C})$ usando la propagazione degli intervalli. La sovra-approssimazione introduce conservatorismo: alcune azioni sicure nel latente vengono classificate come non sicure. Il grado di conservatorismo dipende dalla non-linearità dell'encoder e del decodificatore e può essere ridotto usando approssimazioni più raffinate (zonotopi, poliedri).

5.2 Sicurezza latente con JEPA: il caso del predittore diretto

La struttura della JEPA offre una prospettiva alternativa sulla sicurezza latente che bypassa parzialmente il problema dell'inverso del decodificatore. Nella JEPA gerarchica di LeCun (discussa nel Capitolo III e nelle sezioni applicative del Capitolo IV), la pianificazione avviene interamente nello spazio latente attraverso il modulo predittore H: il sistema non decodifica lo stato futuro nello spazio fisico, ma opera le sue decisioni interamente nel latente.

In questo contesto, la specifica di sicurezza può essere formulata direttamente nel latente senza passare per il decodificatore, a patto di avere una funzione g_latent: $\mathscr{Z} \to \mathbb{R}$ che approssima la specifica fisica $g(\psi(z)) \leq 0$. Questa funzione può essere appresa da esempi di stati sicuri e non sicuri nello spazio latente, usando la stessa pipeline della classificazione delle rappresentazioni. La CBF nel latente ha la forma:

h_latent(z) = -g_latent(z)

e la condizione CBF latente è:

∂h_latent/∂z · (ẑ_{t+1} - z_t)/Δt + α(h_latent(z_t)) ≥ 0

dove ẑ_{t+1} = H(z_t, c_t, a_t) è la predizione del predittore JEPA della rappresentazione al passo successivo in funzione dello stato corrente z_t, del contesto c_t e dell'azione a_t. Il QP sicuro nel latente di JEPA modifica il vettore di azione a_t per garantire la condizione CBF, con una semantica naturalissima: l'agente sceglie l'azione che si avvicina di più al suo obiettivo (u_nom prodotta dal modulo decisionale di LeCun) senza però predire una transizione latente che violerebbe la specifica di sicurezza.

La coerenza tra la CBF latente e la sicurezza fisica dipende dalla qualità dell'encoder: se φ è stabile rispetto a perturbazioni (cioè vicini nello spazio fisico restano vicini nel latente), allora la sicurezza latente implica approssimativamente la sicurezza fisica con un margine proporzionale alla costante di Lipschitz dell'encoder. Questo ricollega la sicurezza alle proprietà geometriche delle rappresentazioni JEPA analizzate nel Capitolo IX, consolidando il filo argomentativo principale della monografia.

6. Composizione delle garanzie: un framework unificato

Le tre linee di ricerca esaminate in questo capitolo — predizione conforme, analisi di raggiungibilità con CBF, verifica formale con shield — non sono alternative ma complementari. La loro composizione forma un framework di sicurezza a livelli che opera su scale temporali diverse e con garanzie di natura diversa.

Al livello più fine — ogni passo di controllo, tempo reale — il *filtro CBF adattivo conforme* (Sezione 3.4) modifica l'azione nominale per garantire l'invarianza positiva dell'insieme sicuro con probabilità 1-α per passo, dove α è calibrato dalla CP sul modello di transizione appreso.

Al livello intermedio — orizzonte di pianificazione del world model, tipicamente H = 10-50 passi — l'*analisi di raggiungibilità con approssimazione neurale dell'HJI* (Sezione 3.2) calcola il backward reachable set sull'orizzonte di pianificazione e lo usa per filtrare le traiettorie immaginarie che violerebbero la specifica di sicurezza prima ancora di essere eseguite.

Al livello più ampio — comportamento a lungo termine della politica — la *sintesi di shield con specifica LTL* (Sezione 4.3) e la *certificazione di Lyapunov* (Sezione 4.1) garantiscono che la politica complessiva rispetti le specifiche di sicurezza strutturali (assenza di collisioni, raggiungimento degli obiettivi, stabilità) indipendentemente dall'orizzonte temporale.

La composizione di queste garanzie si esprime formalmente come segue. Sia P_Lyapunov la probabilità che la politica certificata da Lyapunov sia stabile, sia P_reach(T) la probabilità che il reachable set calcolato sull'HJI non intersechi l'insieme del target entro T passi, e sia P_CBF la probabilità per passo che il filtro CBF conforme sia valido. La probabilità di sicurezza sulla traiettoria di lunghezza T è:

$$P_{safe}(T) \geq P_{Lyapunov} \cdot P_{reach}(T) \cdot P_{CBF}^{T}$$

Con valori tipici $P_{Lyapunov} \approx 1$ (la certificazione è deterministica se completata con successo), $P_{reach}(T) \approx 1-\delta$ (con δ piccolo dalla CP sull'HJI neurale), $P_{CBF} \approx 1-\alpha$ (dalla CP sul modello di transizione), si ottiene:

$$P_{safe}(T) \geq (1-\delta) \cdot (1-\alpha)^{T}$$

che è la garanzia di sicurezza sull'intera traiettoria composta. Per sistemi safety-critical con requisiti normativi (ad esempio ISO 26262 per la guida autonoma), questa formula consente di scegliere α e T in modo da rispettare le soglie di probabilità di guasto accettabili.

7. Il legame con la topologia algebrica e la persistenza della robustezza

Il Capitolo XX di questa monografia introduce la topologia algebrica persistente come strumento per analizzare le rappresentazioni dei world model. La connessione con la sicurezza certificata è più sostanziale di quanto possa sembrare a prima vista e merita di essere sviluppata esplicitamente, colmando uno dei gap strutturali segnalati nell'analisi critica della monografia.

I gruppi di omologia persistente H_k dell'insieme sicuro $\mathscr{C}$ — calcolati filtrandone la topologia per livelli crescenti della funzione vincolo g — catturano le caratteristiche geometriche topologicamente stabili di $\mathscr{C}$: le componenti connesse (H_0), i loop non contraibili (H_1), le cavità chiuse (H_2), eccetera. Queste caratteristiche topologiche sono *robuste* rispetto a perturbazioni della funzione g: se g è perturbata di ε in norma infinito, le caratteristiche topologiche persistono finché la loro *persistenza* (la differenza tra i valori di g alla nascita e alla morte di ciascuna caratteristica) supera 2ε. Questo è il contenuto del teorema di stabilità dei diagrammi di persistenza (Cohen-Steiner et al., 2007).

Nel contesto della sicurezza dei world model, questa robustezza topologica ha un'interpretazione diretta. Quando il world model apprende una rappresentazione approssimata dello spazio degli stati, l'insieme sicuro indotto nel latente $\mathscr{C}$_latent = $\varphi(\mathscr{C})$ è una deformazione topologica dell'insieme sicuro fisico $\mathscr{C}$. Se la mappa encoder φ è ε-vicina all'isometria (condizione che può essere certificata tramite analisi di raggiungibilità sullo spazio dei gradienti dell'encoder), allora le caratteristiche topologiche di $\mathscr{C}$ e $\mathscr{C}$_latent differiscono al più di 2ε in persistenza. Le caratteristiche con alta persistenza — le strutture topologiche robuste di $\mathscr{C}$ — vengono fedelmente trasportate nel latente; le caratteristiche con bassa persistenza — i dettagli geometrici fini — possono scomparire.

Questo risultato suggerisce una strategia di verifica a due livelli: la topologia grossolana dell'insieme sicuro viene verificata nel latente tramite homologia persistente (computazionalmente accessibile con gli algoritmi di Edelsbrunner-Harer); la geometria fine viene verificata nel spazio fisico tramite MILP e propagazione degli intervalli, ma solo per le caratteristiche topologicamente stabili. Le caratteristiche instabili — quelle con bassa persistenza e alta sensibilità alle perturbazioni del modello — vengono trattate come non-verificabili e il sistema adotta una posizione conservativa nei loro confronti. Questa strategia riduce drasticamente il costo computazionale della verifica mantenendo la copertura delle garanzie sulle strutture geometriche che contano davvero per la sicurezza operativa.

Appendice A — Fondamenti Matematici e Convenzioni Notazionali

---

Premessa

Un'opera che attraversa la teoria dell'informazione, la geometria riemanniana, il calcolo stocastico, la topologia algebrica e l'ottimizzazione non lineare non può evitare il problema della coerenza notazionale: ciascuna di queste discipline ha sedimentato nel tempo convenzioni proprie, non sempre compatibili, e la loro giustapposizione in un unico corpo di testo rischia di generare ambiguità sistematiche che compromettono la leggibilità delle argomentazioni formali. Questa appendice non si limita pertanto a raccogliere simboli in un glossario: intende anche *giustificare* le scelte notazionali nei casi in cui esistono alternative consolidate, e derivare le identità fondamentali da cui dipendono i risultati principali del testo. Chi è familiare con l'apprendimento automatico probabilistico troverà in alcune delle sezioni seguenti derivazioni già note; la ragione per includerle è che le versioni del testo principale le presuppongono in forme specifiche che non sempre corrispondono alle presentazioni standard dei manuali.

A.1 Spazi, misure e distribuzioni di probabilità

Lo spazio euclideo $\mathbb{R}^n$ è sempre inteso con la metrica euclidea standard $d(x, y) = \|x - y\|_2$, salvo indicazione esplicita in contrario. Si indica con $\mathbb{R}^+$ la semiretta reale positiva $\{x \in \mathbb{R} : x > 0\}$ e con $\mathbb{R}_0^+$ la sua chiusura $\{x \in \mathbb{R} : x \geq 0\}$. Per spazi di stati, azioni e osservazioni si utilizzano rispettivamente le lettere calligrafiche $\mathcal{S}$, $\mathcal{A}$, $\mathcal{O}$; il simplesso delle distribuzioni di probabilità su un insieme finito $\mathcal{S}$ è indicato con $\Delta(\mathcal{S})$, ovvero $\Delta(\mathcal{S}) = \{p \in \mathbb{R}^{|\mathcal{S}|} : p_i \geq 0 \text{ per ogni } i, \sum_i p_i = 1\}$. Per insiemi continui la distribuzione di probabilità è una misura di probabilità su uno spazio misurabile $(\Omega, \mathcal{F})$, che per semplicità si identifica con la sua densità rispetto alla misura di Lebesgue quando questa esiste.

La distribuzione gaussiana multivariata con vettore medio $\mu \in \mathbb{R}^n$ e matrice di covarianza $\Sigma \in \mathbb{R}^{n \times n}$, simmetrica e semidefinita positiva, è indicata $\mathcal{N}(\mu, \Sigma)$; la sua densità è

$$p(x) = (2\pi)^{-n/2} (\det \Sigma)^{-1/2} \exp\left(-\tfrac{1}{2} (x - \mu)^{\top} \Sigma^{-1} (x - \mu)\right).$$

Quando $\Sigma = \sigma^2 I$ si parla di gaussiana isotropa e si scrive $\mathcal{N}(\mu, \sigma^2 I)$; la gaussiana standard è $\mathcal{N}(0, I)$. La distribuzione categoriale su K categorie con vettore di probabilità $\pi \in \Delta(K)$ è indicata $\mathrm{Cat}(\pi)$; la distribuzione di Bernoulli con parametro $p \in [0,1]$ è $\mathrm{Ber}(p)$. La funzione indicatrice è $\mathbb{1}[\text{condizione}]$, che vale 1 se la condizione è vera e 0 altrimenti.

Per quanto concerne la notazione delle distribuzioni condizionate: $p(x \mid y)$ indica la densità condizionale di X dato $Y = y$; $q_\varphi(z \mid x)$ indica una famiglia di distribuzioni condizionate parametrizzata da φ, cioè un encoder. L'aspettativa di una funzione f rispetto a una distribuzione p è indicata $\mathbb{E}_p[f] = \int f(x)\, p(x)\, dx$, o più brevemente $\mathbb{E}_{x \sim p}[f(x)]$.

A.2 Vettori, matrici e algebra lineare

I vettori sono indicati con lettere minuscole (x, z, h, a, s); le matrici con lettere maiuscole (W, M, A, J). Questa è una scelta contro la convenzione tipografica del grassetto, adottata per semplicità di composizione; il tipo di oggetto (vettore o scalare) è sempre chiaro dal contesto. La trasposta di una matrice A è $A^\top$; l'inversa (quando esiste) è A^{-1}; la pseudo-inversa di Moore-Penrose è A^+.

Le norme vettoriali adoperate con maggiore frequenza sono la norma L^2 euclidea $\|x\|_2 = (\sum_i x_i^2)^{1/2}$, la norma L^1 $\|x\|_1 = \sum_i |x_i|$, e la norma L^∞ $\|x\|_\infty = \max_i |x_i|$. Per le matrici si impiega prevalentemente la norma di Frobenius $\|A\|_F = (\sum_{i} \sum A_{i\square}^2)^{1/2} = (\mathrm{tr}(A^\top A))^{1/2}$, che costituisce la norma indotta dal prodotto interno di Hilbert-Schmidt sullo spazio delle matrici.

Il prodotto interno euclideo tra vettori $x, y \in \mathbb{R}^n$ è $\langle x, y \rangle = x^\top y = \sum_i x_i y_i$; il prodotto di Hadamard (elemento per elemento) è $x \odot y$ con $(x \odot y)_i = x_i y_i$; il prodotto di Kronecker di $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{p \times q}$ è $A \otimes B \in \mathbb{R}^{mp \times nq}$ con $(A \otimes B)_{ij,kl} = A_{i\square} B$. La traccia di una matrice quadrata A è $\mathrm{tr}(A) = \sum_i A_{ii}$; il rango è $\mathrm{rank}(A)$. Il raggio spettrale di A è $\varrho(A) = \max\{|\lambda| : \lambda \in \sigma(A)\}$, dove $\sigma(A)$ è lo spettro (insieme degli autovalori). La matrice identità $n \times n$ è I; il vettore di zeri di dimensione n è 0; la matrice diagonale con il vettore v sulla diagonale è $\mathrm{diag}(v)$.

Una proprietà della traccia che ricorre frequentemente nelle derivazioni: $\mathrm{tr}(AB) = \mathrm{tr}(BA)$ per qualsiasi $A \in \mathbb{R}^{m \times n}$ e $B \in \mathbb{R}^{n \times m}$. Un'identità utile nel calcolo della derivata della log-verosimiglianza gaussiana: per A simmetrica e definita positiva, $\partial / \partial A \log\det A = A^{-1}$.

A.3 Calcolo differenziale e ottimizzazione

Il gradiente di una funzione scalare $f : \mathbb{R}^n \to \mathbb{R}$ rispetto al vettore θ è indicato $\nabla_\theta f$, un vettore colonna di componenti $(\nabla_\theta f)_i = \partial f/\partial\theta_i$. La matrice Jacobiana di una funzione vettoriale $f : \mathbb{R}^n \to \mathbb{R}^m$ è $J_f(x) \in \mathbb{R}^{m\times n}$ con $(J_f)_{i\square} = \partial f_i/\partial x$; la matrice Hessiana di una funzione scalare è $H_f(x) \in \mathbb{R}^{n\times n}$ con $(H_f)_{i\square} = \partial^2 f/\partial x_i \partial x$.

L'operazione di *stop-gradient*, indicata $sg(\cdot)$, è l'identità in avanti ma ha gradiente nullo all'indietro: $sg(x) = x$ ma $\partial/\partial\theta\ sg(f(\theta)) = 0$. Questa operazione è implementata nelle librerie di differenziazione automatica come detach() (PyTorch) o stop_gradient() (JAX) ed è cruciale nell'addestramento di architetture con encoder target come la JEPA.

La regola della catena in forma vettoriale: se $z = f(y)$ e $y = g(x)$, allora $\partial z/\partial x = J_f(y) \cdot J_g(x)$. La derivata rispetto al tempo è indicata con un punto sovrascritto ($\dot{x} = dx/dt$) o con df/dt per funzioni di una singola variabile temporale; le derivate parziali rispetto al tempo si indicano $\partial f/\partial t$.

Nell'ottimizzazione stocastica, $\text{argmin}_\theta f(\theta)$ indica il valore di θ che minimizza f; quando la funzione ha più minimi locali, si intende un minimo locale raggiunto dall'algoritmo di ottimizzazione. L'aggiornamento del gradiente stocastico con learning rate η è $\theta \leftarrow \theta - \eta\ \nabla_\theta \mathcal{L}(\theta; x, y)$. *L'ottimizzatore Adam mantiene stime del primo momento* $m_t = \beta_1\, m_{t-1} + (1 - \beta_1)\, g_t$ e del secondo momento $v_t = \beta_2\, v_{t-1} + (1 - \beta_2)\, g_t^2$ delle stime del gradiente g_t, con correzione del bias $\hat{m}_t = m_t/(1 - \beta_1^t)$, $\hat{v}_t = v_t/(1 - \beta_2^t)$, e aggiornamento $\theta_{t+1} = \theta_t - \eta\ \hat{m}_t / (\sqrt{\hat{v}_t} + \varepsilon)$. L'aggiornamento a media mobile esponenziale (EMA) con momento m è $\text{EMA}(\theta, m) = m \cdot \theta_{\text{old}} + (1 - m) \cdot \theta_{\text{new}}$, e si impiega sistematicamente per aggiornare l'encoder target nelle architetture JEPA.

A.4 Teoria dell'informazione: derivazioni fondamentali

L'entropia di Shannon di una variabile aleatoria discreta X con distribuzione p è $H(X) = -\Sigma_x\, p(x) \log p(x)$; per una variabile continua con densità p è $H(X) = -\int p(x) \log p(x)\, dx$ (entropia differenziale). Il logaritmo è in base e (nats) salvo indicazione contraria. L'entropia condizionale è $H(X \mid Y) = -\Sigma_\gamma\, p(y)\ \Sigma_x\, p(x|y) \log p(x|y) = H(X, Y) - H(Y)$. L'informazione mutua è $I(X; Y) = H(X) - H(X \mid Y) = H(Y) - H(Y \mid X) = \text{KL}[p(x, y) \,\|\, p(x)p(y)]$, dove KL denota la divergenza di Kullback-Leibler.

La *divergenza di Kullback-Leibler* tra distribuzioni p e q è definita da

$$\text{KL}[p \,\|\, q] = \int p(x) \log(p(x)/q(x))\, dx,$$

e soddisfa $\text{KL}[p \,\|\, q] \geq 0$ con uguaglianza se e solo se $p = q$ quasi ovunque (disuguaglianza di Gibbs, immediata dalla concavità del logaritmo e dalla disuguaglianza di Jensen). È importante notare che $\text{KL}[p \,\|\, q] \neq \text{KL}[q \,\|\, p]$ in generale:

la versione *forward* KL[p ‖ q] penalizza le regioni dove p è grande ma q è piccola (comportamento *zero-forcing* per q), mentre la versione *reverse* KL[q ‖ p] penalizza le regioni dove p è piccola ma q è grande (comportamento *mass-covering*). Nella loss dell'ELBO compare la versione reverse KL[$q_\varphi(z|x)$ ‖ $p(z|x)$], che in fase di addestramento tende a formare approssimazioni variazionali che evitano le regioni dove la vera posteriore ha massa bassa.

La *divergenza di Jensen-Shannon* è la simmetrizzazione JS[p ‖ q] = ½ KL[p ‖ m] + ½ KL[q ‖ m] dove m = (p + q)/2; essa è simmetrica, limitata in [0, log 2] e la sua radice quadrata è una metrica (distanza di Jensen-Shannon).

Per la gaussiana, la divergenza KL ha forma chiusa. Tra due gaussiane univariate $\mathcal{N}(\mu_1, \sigma_1^2)$ e $\mathcal{N}(\mu_2, \sigma_2^2)$:

$$\mathrm{KL}[\mathcal{N}(\mu_1, \sigma_1^2) \,\|\, \mathcal{N}(\mu_2, \sigma_2^2)] = \log(\sigma_2/\sigma_1) + (\sigma_1^2 + (\mu_1 - \mu_2)^2)/(2\sigma_2^2) - \tfrac{1}{2}.$$

Per gaussiane multidimensionali $\mathcal{N}(\mu_1, \Sigma_1)$ e $\mathcal{N}(\mu_2, \Sigma_2)$ con Σ_2 invertibile:

$$\mathrm{KL}[\mathcal{N}(\mu_1, \Sigma_1) \,\|\, \mathcal{N}(\mu_2, \Sigma_2)] = \tfrac{1}{2}\,[\mathrm{tr}(\Sigma_2^{-1}\Sigma_1) + (\mu_2 - \mu_1)^{\mathsf{T}}\,\Sigma_2^{-1}\,(\mu_2 - \mu_1) - n + \log(\det \Sigma_2 / \det \Sigma_1)],$$

dove n è la dimensione. Caso speciale essenziale nel calcolo dell'ELBO: divergenza KL tra un encoder gaussiano $q_\varphi(z|x) = \mathcal{N}(z; \mu_\varphi(x), \mathrm{diag}(\sigma_\varphi^2(x)))$ e il prior standard $p(z) = \mathcal{N}(0, I_d)$:

$$\mathrm{KL}[\mathcal{N}(\mu, \mathrm{diag}(\sigma^2)) \,\|\, \mathcal{N}(0, I)] = \tfrac{1}{2} \sum (\mu^2 + \sigma^2 - \log \sigma^2 - 1),$$

dove la somma scorre sulle d dimensioni latenti. Questa formula è la versione analitica del termine di regolarizzazione nei VAE e viene impiegata ad ogni passo di addestramento senza la necessità di campionamento Monte Carlo.

A.5 Derivazione completa dell'ELBO e del reparametrization trick

Si consideri un modello generativo $p_\theta(x, z) = p_\theta(x|z)\, p(z)$ dove $x \in \mathcal{X}$ è l'osservazione, $z \in \mathbb{R}^d$ è la variabile latente, e $p(z) = \mathcal{N}(0, I_d)$ è il prior. L'obiettivo dell'apprendimento per massima verosimiglianza marginale è

$$\max_\theta \log p_\theta(x) = \max_\theta \log \int p_\theta(x|z)\, p(z)\, dz.$$

Questo integrale è intrattabile in alta dimensione. Si introduce una famiglia di distribuzioni variazionali $q_\varphi(z|x)$ (l'encoder) e si scrive, per qualunque scelta di q_φ:

$$\log p_\theta(x) = \log \int q_\varphi(z|x) \cdot [p_\theta(x|z)\, p(z) / q_\varphi(z|x)]\, dz \geq \mathbb{E}_{q_\varphi(z|x)} [\log (p_\theta(x|z)\, p(z) / q_\varphi(z|x))] \text{ (Jensen)} = \mathbb{E}_{q_\varphi(z|x)} [\log p_\theta(x|z)] - \mathrm{KL}[q_\varphi(z|x) \,\|\, p(z)] =: \mathcal{L}(\theta, \varphi; x) \text{ (ELBO)}.$$

La disuguaglianza di Jensen è applicabile perché il logaritmo è una funzione concava. L'esatta quantità persa nella disuguaglianza è

$$\log p_\theta(x) - \mathcal{L}(\theta, \varphi; x) = KL[q_\varphi(z|x) \| p_\theta(z|x)] \geq 0,$$

come si verifica espandendo la KL e riconoscendo la normalizzazione di $p_\theta(z|x) = p_\theta(x|z)p(z)/p_\theta(x)$.

L'ottimizzazione dell'ELBO richiede il calcolo del gradiente $\nabla_\varphi \mathcal{L}(\theta, \varphi; x)$. *Il termine* $-KL[q_\varphi \| p]$ *è differenziabile rispetto a* φ *in forma chiusa (se le distribuzioni sono gaussiane). Il termine* $\mathbb{E}_{q_\varphi(z|x)}[\log p_\theta(x|z)]$ richiede invece una stima del gradiente di un'aspettazione rispetto a una distribuzione che dipende dai parametri stessi.

Il *reparametrization trick* risolve questa difficoltà parametrizzando il campionamento dalla distribuzione q_φ come funzione differenziabile di una variabile ausiliaria ε con distribuzione fissa. Per $q_\varphi(z|x) = \mathcal{N}(z; \mu_\varphi(x), \mathrm{diag}(\sigma_\varphi^2(x)))$:

$$z = \mu_\varphi(x) + \sigma_\varphi(x) \odot \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, I_d).$$

In questo modo z è una funzione deterministica e differenziabile di $(x, \varphi, \varepsilon)$, e il gradiente rispetto a φ può fluire attraverso la funzione $z(\cdot; \varphi, \varepsilon)$. Lo stimatore del gradiente risultante,

$$\nabla_\varphi \mathbb{E}_{q_\varphi(z|x)}[f(z)] \approx \nabla_\varphi f(\mu_\varphi(x) + \sigma_\varphi(x) \odot \varepsilon) \text{ (campionato da } \varepsilon \sim \mathcal{N}(0,I)),$$

ha varianza tipicamente molto inferiore agli stimatori REINFORCE (log-derivative) alternativi, il che lo rende il metodo di scelta per i VAE e per i world model che ereditano la loro struttura.

A.6 Notazione per le architetture di rete

I parametri di una rete neurale sono indicati con lettere greche, scelte in modo coerente attraverso il testo: θ denota i parametri del decoder (o del contesto encoder in I-JEPA), φ quelli dell'encoder (o del posterior nel RSSM), ψ quelli di un encoder target o di un modello di ricompensa, ω quelli di una policy. Una funzione neurale parametrizzata da θ è indicata f_θ; quando si vuole enfatizzare la struttura dell'architettura (ad esempio nel meta-learning), si scrive $f(\cdot; \theta)$.

Le dimensioni ricorrenti sono d per la dimensione dello spazio latente, d_k e d_v per dimensioni di chiavi e valori nei Transformer, N per il numero di layer, T per la lunghezza di una sequenza, H per l'orizzonte di pianificazione. I learning rate sono indicati η (globale) o α (inner loop nel meta-learning), in linea con la convenzione MAML. Il parametro di sconto nel reinforcement learning è γ; il parametro λ controlla il trade-off bias-varianza nello stimatore λ-ritorno.

Le funzioni di attivazione principali sono: la sigmoide $\sigma(x) = 1/(1 + e^{-x})$ con proprietà $\sigma'(x) = \sigma(x)(1 - \sigma(x))$; la softmax, $\text{softmax}(x)_i = \exp(x_i) / \Sigma \exp(x)$, con la proprietà che $\partial \text{softmax}(x)_i / \partial x = \text{softmax}(x)_i (\delta_i - \text{softmax}(x))$; la ReLU, $\text{ReLU}(x) = \max(0, x)$; e la GELU, $\text{GELU}(x) = x \cdot \Phi(x)$ dove Φ è la funzione di distribuzione cumulativa gaussiana standard. La layer normalization LN normalizza i vettori di attivazione lungo la dimensione delle feature: $\text{LN}(x) = (x - \mu_x) / \sigma_x \cdot \gamma + \beta$, dove μ_x e σ_x sono media e deviazione standard calcolate sulle componenti di x, mentre γ e β sono parametri appresi.

A.7 Teoria dell'informazione di Fisher e geometria dell'informazione

La *matrice di informazione di Fisher* di una famiglia statistica parametrica $\{p_\theta : \theta \in \Theta\}$ è definita da

$$F(\theta)_{ij} = \mathbb{E}_{p_\theta}\left[\partial_i \log p_\theta(x) \cdot \partial_j \log p_\theta(x)\right] = -\mathbb{E}_{p_\theta}\left[\partial_i \partial_j \log p_\theta(x)\right],$$

dove $\partial_i = \partial/\partial\theta_i$. L'uguaglianza tra le due espressioni vale sotto condizioni di regolarità (interscambiabilità di derivazione e integrazione). $F(\theta)$ è una matrice semidefinita positiva che funge da tensore metrico riemanniano sullo spazio dei parametri — questa è la struttura fondamentale dell'*information geometry* di Amari. Il *gradiente naturale* è $\tilde{\nabla} f(\theta) = F(\theta)^{-1} \nabla f(\theta)$, che aggiorna i parametri nella direzione di massima crescita relativa alla metrica di Fisher piuttosto che alla metrica euclidea standard. Questa scelta è invariante rispetto alla riparametrizzazione del modello — un'importante proprietà teorica di cui i metodi del gradiente ordinario sono privi.

La connessione con la divergenza KL è esplicita: per variazioni infinitesime $d\theta$,

$$\mathrm{KL}[p_\theta \,\|\, p_{\theta + d\theta}] = \tfrac{1}{2}\, d\theta^{T} F(\theta)\, d\theta + O(\|d\theta\|^3),$$

quindi $F(\theta)$ descrive la curvatura del paesaggio KL nelle vicinanze di θ. Questo è il fondamento teorico dell'interpretazione della F come metrica naturale sullo spazio dei parametri.

A.8 Geometria differenziale: varietà, connessioni e curvatura

Una *varietà differenziabile* $\mathcal{M}$ di dimensione n è uno spazio topologico Hausdorff, a base numerabile, dotato di atlante $\{(U_\alpha, \varphi_\alpha)\}$, dove ciascuna U_α è un aperto di $\mathcal{M}$ e $\varphi_\alpha : U_\alpha \to \mathbb{R}^n$ è un omeomorfismo tale che le mappe di transizione $\varphi_\beta \circ \varphi_\alpha^{-1}$ siano differenziabili. Lo *spazio tangente* in $p \in \mathcal{M}$, indicato $T_p\mathcal{M}$, è lo spazio vettoriale delle classi di equivalenza di curve lisce passanti per p con la stessa tangente. Una *varietà*

riemanniana $(\mathcal{M}, g)$ è una varietà dotata di tensore metrico g, cioè una sezione liscia del fibrato dei tensori simmetrici semidefiniti positivi di tipo (0,2): ad ogni $p \in \mathcal{M}$, g_p è un prodotto interno su $T_p \mathcal{M}$.

I *simboli di Christoffel* della connessione di Levi-Civita associata a g sono

$$\Gamma^k_{ij} = \tfrac{1}{2}\, g^{kl} \left(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}\right),$$

dove g^{kl} sono le componenti del tensore metrico inverso. La connessione di Levi-Civita è l'unica connessione compatibile con g ($\nabla g = 0$) e priva di torsione. Il *trasporto parallelo* di un vettore V lungo una curva $\gamma(t)$ è la soluzione di $dV^k/dt + \Gamma^k_{ij} (d\gamma^i/dt)\, V^j = 0$.

Il *tensore di curvatura di Riemann* è $R^l_{ijk} = \partial_j \Gamma^l_{ik} - \partial_k \Gamma^l_{ij} + \Gamma^l_{jm} \Gamma^m_{ik} - \Gamma^l_{km} \Gamma^m_{ij}$; la curvatura sezionale $K(u, v) = g(R(u,v)v, u) \,/\, (g(u,u)g(v,v) - g(u,v)^2)$ misura la curvatura del piano generato da $u, v \in T_p \mathcal{M}$. Lo spazio euclideo ha curvatura identicamente nulla; la sfera unitaria S^n ha curvatura sezionale $+1$; lo spazio iperbolico $\mathbb{H}^n$ ha curvatura sezionale -1. Queste due ultime geometrie sono rilevanti per la monografia: la curvatura positiva caratterizza le rappresentazioni sferiche (adatte a spazi di probabilità compatti), quella negativa le rappresentazioni iperboliche (adatte a strutture gerarchiche, poiché lo spazio iperbolico n-dimensionale può incorporare isometricamente alberi con distorsione arbitrariamente piccola).

A.9 Topologia algebrica: omologia e persistenza

Il *complesso simpliciale* K su un insieme di vertici V è una famiglia di sottoinsiemi di V (i *simplessi*) chiusa per inclusione: se $\sigma \in K$ e $\tau \subseteq \sigma$ allora $\tau \in K$. Un q-simplesso $\sigma = [v_0, \ldots, v_q]$ è un sottoinsieme di q+1 vertici; la sua *frontiera* è $\partial\sigma = \sum_i (-1)^i [v_0, \ldots, \hat{v_i}, \ldots, v_q]$, dove il cappello indica omissione. Il *q-esimo gruppo di catene* $C_q(K; \mathbb{F})$ è lo spazio vettoriale su $\mathbb{F}$ generato dai q-simplessi di K. L'operatore di frontiera $\partial_q : C_q \to C_{q-1}$ soddisfa $\partial_{q-1} \circ \partial_q = 0$, il che permette di definire

$$H_q(K; \mathbb{F}) = \ker \partial_q \,/\, \mathrm{im}\, \partial_{q+1}$$

come *q-esimo gruppo di omologia* di K a coefficienti in $\mathbb{F}$. Il q-esimo *numero di Betti* $\beta_q = \operatorname{rank} H_q(K; \mathbb{Q})$ conta le cavità q-dimensionali: β_0 è il numero di componenti connesse, β_1 è il numero di cicli indipendenti, β_2 è il numero di vuoti tridimensionali, e così via.

L'*analisi persistente* (persistent homology) estende questo calcolo a funzioni di filtraggio: data $f : V \to \mathbb{R}$, si costruisce la *filtrazione* $K_{t_1} \subseteq K_{t_2} \subseteq \ldots \subseteq K_{t_n} = K$ aggiungendo simplessi nell'ordine dato dai valori di f sui vertici e sugli spigoli (complesso di Vietoris-Rips o di Čech). Per ogni grado q, si traccia la *nascita* b e la *morte* d di ogni classe di omologia al crescere del parametro di filtraggio, ottenendo il *diagramma di persistenza* $\mathrm{Dgm}_q(K, f)$ come insieme di punti (b, d) nel piano con $d \geq b$.

La *lunghezza di vita* $d - b$ misura la prominenza della caratteristica topologica. Le distanze tra diagrammi di persistenza impiegate nel testo sono la distanza di bottleneck $d_B(D_1, D_2) = \inf_{\gamma \text{ biiezione}} \sup_{p \in D_1} \|p - \gamma(p)\|_\infty$ e la distanza di Wasserstein $d_{W^p}(D_1, D_2) = (\inf_\gamma \sum_p \|p - \gamma(p)\|^p)^{1/p}$, entrambe metriche stabili rispetto a perturbazioni della funzione di filtraggio.

A.10 Meccanica hamiltoniana e struttura simplettica

Lo spazio delle fasi di un sistema meccanico con n gradi di libertà è $\mathbb{R}^{2n}$ con coordinate (q, p) (posizioni e momenti generalizzati). La *2-forma simplettica* standard è $\omega = \sum_i dp_i \wedge dq_i$; in forma matriciale, la struttura simplettica è codificata dalla matrice antisimmetrica $J = \begin{bmatrix} 0 & I \\ -I & 0 \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$. Un'applicazione $\Phi : \mathbb{R}^{2n} \to \mathbb{R}^{2n}$ è *simplettica* (o *canonica*) se preserva ω, ovvero se $\Phi^*\omega = \omega$, equivalentemente se $J_\Phi^{\mathsf{T}} J J_\Phi = J$, dove J_Φ è la matrice Jacobiana di Φ.

Le *equazioni di Hamilton* per un sistema con hamiltoniana $H(q, p)$ sono

$$\dot{q}_i = \frac{\partial H}{\partial p_i}, \quad \dot{p}_i = -\frac{\partial H}{\partial q_i},$$

o in forma compatta $\dot{z} = J \nabla H(z)$ dove $z = (q, p)^{\mathsf{T}}$. Il flusso hamiltoniano preserva l'hamiltoniana (H è una costante del moto: $dH/dt = \nabla H \cdot \dot{z} = \nabla H \cdot J \nabla H = 0$ per antisimmetria di J) e la misura di Liouville ($\det J_\Phi = 1$, teorema di Liouville). Il metodo simplettico di *Störmer-Verlet* integra numericamente le equazioni di Hamilton con passo h:

$$p_{n+1/2} = p_n - \tfrac{h}{2} \nabla_q H(q_n, p_{n+1/2}), \quad q_{n+1} = q_n + h \nabla_p H(q_n, p_{n+1/2}),$$
$$p_{n+1} = p_{n+1/2} - \tfrac{h}{2} \nabla_q H(q_{n+1}, p_{n+1/2}).$$

Questo integratore è di secondo ordine in h, simmetrico (reversibile nel tempo), e preserva la struttura simplettica esattamente (è un'applicazione simplettica esatta), il che lo rende superiore agli integratori standard di Runge-Kutta per le simulazioni a lungo termine di sistemi conservativi.

A.11 Processi stocastici e calcolo di Itô

Un *processo di Wiener standard* W_t è un processo stocastico con $W_0 = 0$, incrementi $W_t - W_s \sim \mathcal{N}(0, t - s)$ per $t > s \geq 0$, e incrementi indipendenti su intervalli disgiunti. Una *equazione differenziale stocastica* (SDE) nella forma di Itô è

$$dX_t = f(X_t, t)\, dt + g(X_t, t)\, dW_t,$$

dove f è il drift e g è il coefficiente di diffusione. La formula di Itô per una funzione U(X_t, t) sufficientemente regolare è

$$dU = (\partial_t U + f \cdot \nabla_x U + \tfrac{1}{2} \operatorname{tr}(g\, g^{T} H_U))\, dt + (\nabla_x U)^{T} g\, dW_t,$$

dove H_U è la matrice Hessiana di U rispetto a x. Questa formula si distingue dalla regola della catena ordinaria per il termine ½ tr(g g^T H_U), la *correzione di Itô*, che emerge dalla non-derivabilità quasi-ovunque dei percorsi browniani. La formulazione di Stratonovich (alternativa alla forma di Itô) non ha questa correzione aggiuntiva ma è meno comoda per il calcolo del coefficiente di diffusione e per la relazione con le equazioni di Fokker-Planck.

La *equazione di Fokker-Planck* associata all'SDE di Itô descrive l'evoluzione della densità di probabilità ϱ(x, t) degli stati:

$$\partial_t \varrho = -\nabla \cdot (f \varrho) + \tfrac{1}{2} \Sigma_{i} \partial_i \partial (D_i \varrho),$$

dove D = g g^T è il tensore di diffusione. All'equilibrio (*$\partial t\, \varrho = 0$), per un sistema con energia potenziale E(x), la densità stazionaria è la distribuzione di Boltzmann* $\varrho^{\infty}(x) \propto \exp(-E(x)/T)$, dove T è la temperatura effettiva del rumore.

A.12 Note sulle convenzioni tipografiche

Nei capitoli del testo principale, le formule sono presentate con simboli Unicode allineati alla notazione qui definita; i pedici e gli apici temporali sono a pedice (s_{t+1}, non s(t+1)) per conformità con la notazione RSSM; gli operatori matematici non alfabetici (∇, ∂, $\sum$, $\prod$, $\int$, $\otimes$, $\odot$, $\|\cdot\|$) hanno sempre il medesimo significato a prescindere dal capitolo di occorrenza. Le derivazioni matematiche sono tipograficamente distinte dal testo discorsivo mediante rientro; le definizioni nominali sono introdotte con il costrutto "si definisce ... come" seguito dalla formula, senza uso della notazione "Def." o di riquadri tipografici.

Appendice B — Analisi degli Algoritmi Fondamentali

Premessa

La tradizione dell'informatica teorica insegna a distinguere la *specifica* di un algoritmo — che cosa calcola — dalla sua *analisi* — perché lo calcola correttamente, con quale efficienza, e sotto quali condizioni le garanzie teoriche si trasferiscono alla pratica. Nei

manuali di apprendimento automatico questa distinzione è spesso sacrificata alla convenienza didattica: si presenta lo pseudocodice, si discutono informalmente i motivi delle scelte implementative, e ci si affida alla validazione empirica per la garanzia di correttezza. Questa appendice propone un approccio diverso: ciascuno degli algoritmi centrali della monografia è trattato come un oggetto matematico, di cui si derivano le proprietà analitiche, si individuano le condizioni di convergenza, si precisa la complessità computazionale e si identificano i punti di vulnerabilità — ossia le assunzioni che, quando violate, degradano le prestazioni o invalidano le garanzie teoriche.

L'ordine di presentazione segue la progressione argomentativa del testo principale: si apre con la procedura di addestramento end-to-end di DreamerV3 come archetipo del world model generativo, si passa all'algoritmo di preaddestramento I-JEPA come rappresentante della famiglia delle architetture predittive latenti, si analizzano le proprietà del meta-gradiente in MAML, si sviluppa la pianificazione MCTS nello spazio latente di MuZero, e infine si trattano FlashAttention-2, SAM e MPPI come algoritmi di supporto computazionale e di controllo.

B.1 DreamerV3: addestramento end-to-end del world model

Struttura del problema di ottimizzazione

DreamerV3 ottimizza congiuntamente il world model (WM), composto da encoder q_φ, modello di transizione f_φ, decoder p_θ e modello di ricompensa p_ψ, e un sistema attore-critico (π_ω, V_ξ). L'obiettivo complessivo è la massimizzazione del ritorno atteso scontato

$$J(\omega) = \mathbb{E}_{\tau \sim \pi_\omega} \left[\sum_{t=0}^{\infty} \gamma^t r_t\right],$$

perseguita indirettamente attraverso la sequenza: (i) addestramento del WM su traiettorie reali, (ii) generazione di rollout immaginari tramite il WM, (iii) addestramento di π_ω e V_ξ sui rollout immaginari.

Loss del world model e sua derivazione

La loss del WM è l'ELBO sulla sequenza di lunghezza T campionata dal replay buffer:

$$\mathcal{L}_{WM}(\theta, \varphi, \psi) = \sum_{t=1}^{T} \Big\{ \mathbb{E}_{q_\varphi(z_t \mid h_t, o_t)} \big[\log p_\theta(o_t \mid h_t, z_t) + \log p_\psi(r_t \mid h_t, z_t)\big] - \beta \cdot \mathrm{KL}_{\mathrm{sym}}\big(q_\varphi(z_t \mid h_t, o_t) \,\|\, p_\varphi(z_t \mid h_t)\big) \Big\},$$

dove $h_t = f_\varphi(h_{t-1}, z_{t-1}, a_{t-1})$ è lo stato deterministico calcolato ricorsivamente e $\beta > 0$ è un peso di regolarizzazione. La *KL simmetrica* di DreamerV3 è

$$\mathrm{KL}_{\mathrm{sym}}(q \,\|\, p) = \alpha \cdot \mathrm{KL}[\mathrm{sg}(q) \,\|\, p] + (1 - \alpha) \cdot \mathrm{KL}[q \,\|\, \mathrm{sg}(p)],$$

con $\alpha = 0.8$. Il termine $\mathrm{sg}(\cdot)$ blocca il gradiente: il primo addendo aggiorna solo il prior p_φ (il "rappresentante del futuro immaginato"), il secondo aggiorna solo il posterior q_φ (il "rappresentante della percezione corrente"). Questa asimmetria stabilizza l'addestramento perché evita che le due distribuzioni si inseguano simultaneamente — un fenomeno noto come *gradient interference* nelle reti con cicli di feedback.

Proprietà della GRU e propagazione del gradiente

Lo stato deterministico h_t è prodotto da una Gated Recurrent Unit (GRU), la cui struttura è

$r_t = \sigma(W_r [h_{t-1}; x_t] + b_r)$ (gate di reset), $u_t = \sigma(W_u [h_{t-1}; x_t] + b_u)$ (gate di aggiornamento), $c_t = \tanh(W_c [r_t \odot h_{t-1}; x_t] + b_c)$ (candidato), $h_t = (1 - u_t) \odot h_{t-1} + u_t \odot c_t$,

dove $x_t = [z_{t-1}; a_{t-1}]$ è l'input all'istante t. La motivazione analitica della GRU rispetto all'LSTM o alla cella ricorrente semplice risiede nel compromesso tra capacità espressiva e stabilità del gradiente: il gate di aggiornamento u_t controlla quanto dello stato precedente viene conservato, e quando $u_t \approx 0$ lo stato $h_t \approx h_{t-1}$ implica che il gradiente fluisce indisturbato attraverso quell'istante temporale (gradinete che non svanisce né esplode localmente). La backpropagation through time (BPTT) su T passi produce un termine

$$\partial h_T / \partial h_t = \Pi_{k=t}^{T-1} \partial h_{k+1} / \partial h_k,$$

e la stabilità numerica richiede che il raggio spettrale del prodotto di Jacobiane sia controllato. Con la GRU, quando $u_k \approx 0$ la Jacobiana $\partial h_{k+1} / \partial h_k \approx I$, garantendo un flusso del gradiente vicino all'identità. La normalizzazione dei layer applicata prima della cella GRU stabilizza ulteriormente l'addestramento.

λ-ritorno e stimatore del valore

Il λ-ritorno è definito ricorsivamente come $V_\tau^\lambda = r_{\tau+1} + \gamma[(1 - \lambda) V_\xi(h_{\tau+1}, z_{\tau+1}) + \lambda V_{\tau+1}^\lambda]$, che si riscrive in forma chiusa come

$$V_\tau^\lambda = (1 - \lambda) \Sigma_{n=1}^{H-\tau-1} \lambda^{n-1} G_\tau^n + \lambda^{H-\tau-1} G_\tau^{H-\tau},$$

dove $G_\tau^n = \Sigma_{k=1}^{n} \gamma^{k-1} r_{\tau+k} + \gamma^n V_\xi(h_{\tau+n}, z_{\tau+n})$ è il ritorno n-passi. Per $\lambda = 0$ si recupera il target TD(0); per $\lambda = 1$ il ritorno Monte Carlo; per $0 < \lambda < 1$ si ottiene un interpolazione che bilancia il bias (basso per $\lambda \to 1$) con la varianza (bassa per $\lambda \to 0$). DreamerV3 usa tipicamente $\lambda = 0.95$.

Complessità computazionale

Per un episodio di T passi, il forward pass del WM ha complessità $O(T \cdot d^2)$ per la GRU (dove d è la dimensione dello stato) più $O(T \cdot C_{dec})$ per il decoder, dove C_{dec} dipende dall'architettura del decoder (convoluzionale per input visivi, lineare per input vettoriali). Il rollout immaginario di lunghezza H su B traiettorie ha

complessità $O(B \cdot H \cdot d^2)$. Il training dell'attore via gradiente analitico attraverso i rollout immaginari richiede BPTT attraverso H passi, con complessità $O(B \cdot H \cdot d^2 \cdot |\omega|)$ dove $|\omega|$ è il numero di parametri della policy.

B.2 I-JEPA: analisi della loss e delle proprietà anti-collasso

Formulazione del problema di ottimizzazione

I-JEPA ottimizza congiuntamente l'encoder di contesto f_θ e il predittore h_φ, mantenendo un encoder target g_ψ aggiornato tramite EMA. L'obiettivo è

$$\min_{\theta, \varphi} \mathbb{E}_{x, M_{ctx}, \{y_k\}} \left[\frac{1}{K} \sum_k \frac{1}{|y_k|} \sum_{p \in y_k} \left\| \hat{h}_{y^p} / \|\hat{h}_{y^p}\|_2 - s_{y^p} / \|s_{y^p}\|_2 \right\|_2^2 \right],$$

dove $s_x = f_\theta(x_{ctx})$ è la rappresentazione del contesto, $\hat{s}_{y_k} = h_\varphi(s_x, pos_{y_k})$ è la predizione del predittore condizionata sulla posizione delle regioni target, e $s_{y_k} = sg(g_\psi(x))$ sono i target ottenuti dall'encoder target su cui non scorrono gradienti.

Il meccanismo anti-collasso: analisi formale

Il pericolo del collasso delle rappresentazioni si manifesta in due forme distinte che richiedono meccanismi di difesa diversi.

La prima forma è il *collasso completo*: f_θ collassa tutte le rappresentazioni verso un singolo punto o una regione degenere, minimizzando la loss indipendentemente dall'input. Questo è prevenuto dall'asimmetria architetturale: il predittore h_φ è condizionato sulla posizione della regione target, e deve quindi *generalizzare* dalla rappresentazione del contesto alla posizione corretta, il che richiede che f_θ mantenga informazione spazialmente differenziata.

La seconda forma è il *collasso dimensionale*: le rappresentazioni si concentrano su un sottospazio di dimensione bassa, perdendo capacità rappresentativa. La normalizzazione L2 nelle rappresentazioni target ($s_{y^p} / \|s_{y^p}\|_2$) proietta le rappresentazioni sulla sfera unitaria, impedendo che il modello minimizzi la loss semplicemente scalando le rappresentazioni verso zero. Tuttavia, rappresentazioni che collassano su una singola direzione sulla sfera unitaria avrebbero norma 1 ma rango 1, quindi il meccanismo di normalizzazione da solo non impedisce questa forma di degenerazione.

L'architettura I-JEPA si affida a una combinazione di tre fattori per evitare il collasso dimensionale: (i) la capacità limitata del predittore h_φ rispetto all'encoder f_θ, che impedisce soluzioni triviali in cui il predittore semplicemente copia la rappresentazione

del contesto; (ii) il masking della regione di contesto, che introduce una discrepanza strutturale tra l'input di f_θ e l'input di g_ψ, forzando f_θ ad apprendere rappresentazioni che sono *predittivamente utili* e non semplicemente ricche di informazione locale; (iii) l'aggiornamento EMA dell'encoder target, che introduce una forma di regolarizzazione temporale: g_ψ si avvicina lentamente a f_θ, evitando oscillazioni e mantenendo una separazione temporale tra il punto fisso dell'ottimizzazione (g_ψ) e la variabile ottimizzata (f_θ).

Relazione con la teoria dell'informazione

Si può dimostrare che, sotto condizioni di regolarità, minimizzare la loss di I-JEPA equivale a massimizzare un limite inferiore alla stima di condivisione di informazione tra contesto e target nello spazio latente. Formalmente, sia $Z_{ctx} = f_\theta(X_{ctx})$ e $Z_{tgt} = g_\psi(X_{tgt})$; allora la loss di I-JEPA è approssimativamente

$$\mathcal{L} \approx 2 - 2 \cdot \mathbb{E}\left[\left\langle Z_{ctx} / \|Z_{ctx}\|, Z_{tgt} / \|Z_{tgt}\| \right\rangle\right] = 2 - 2 \cdot \cos(Z_{ctx}, Z_{tgt}).$$

Massimizzare $\cos(Z_{ctx}, Z_{tgt})$ è equivalente a massimizzare la distanza coseno, che è una proxy dell'informazione mutua normalizzata. La struttura di masking garantisce che questa informazione mutua riguardi le strutture semantiche di alto livello — quelle che sopravvivono alla rimozione del contesto locale — piuttosto che i dettagli pixel-level.

Complessità e scelta del masking

La complessità computazionale di un forward pass di I-JEPA su un'immagine suddivisa in P patch è $O(P^2 \cdot d)$ per l'encoder basato su ViT (grazie alla self-attention quadratica), ridotta a $O(|ctx|^2 \cdot d)$ quando si applica l'attention solo alle patch del contesto. Il costo del predittore è $O(|ctx|^2 + |ctx| \cdot \Sigma_k |y_k|)$ per la cross-attention tra patch di contesto e patch target. La scelta ottimale della dimensione del contesto — tipicamente tra il 10% e il 25% dell'immagine totale — bilancia la difficoltà del compito di predizione (più contesto rende la predizione più facile) con la ricchezza della supervisione (più contesto riduce la varianza dello stimatore del gradiente).

B.3 MAML: analisi del meta-gradiente e approssimazioni

Il meta-gradiente esatto e il suo costo

L'aggiornamento esterno di MAML richiede il calcolo di

$$\nabla_\Theta \Sigma_i \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) = \Sigma_i J_{\{\theta'_i \to \Theta\}}^T \nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i},$$

dove $J_{\theta'_i \to \Theta}$ è la Jacobiana dell'adattamento interno $\theta'_i = \Theta - \alpha \nabla_\Theta \mathcal{L}_{\mathcal{T}_i}^{sup}(f_\Theta)$ rispetto a Θ. Si ha

$$\partial \theta'_i / \partial \Theta = I - \alpha \, \partial^2 \mathcal{L}_{\mathcal{T}_i}^{sup}(f_\Theta) / \partial \Theta^2,$$

ovvero la Jacobiana dell'adattamento interno coinvolge la matrice Hessiana della loss di supporto. Il calcolo esatto del meta-gradiente richiede dunque la valutazione di prodotti Hessiani vettoriali, un'operazione di complessità $O(|\Theta|)$ per forward-mode autodiff o $O(|\Theta|)$ con il trucco di Pearlmutter — ma che richiede un doppio passaggio di backpropagation, raddoppiando il costo computazionale e la memoria richiesta rispetto al gradiente di primo ordine.

FOMAML: l'approssimazione di primo ordine e il suo errore

L'approssimazione di primo ordine (FOMAML) ignora il termine Hessiano e approssima

$$\nabla_\Theta \Sigma_i \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i}) \approx \Sigma_i \nabla_{\theta'_i} \mathcal{L}_{\mathcal{T}_i},$$

trattando θ'_i come se fosse indipendente da Θ. L'errore di questa approssimazione è di ordine $O(\alpha \, \|H\|)$, dove H è la norma della Hessiana della loss di supporto. Nei regimi pratici con α piccolo e funzioni di loss ragionevolmente condizionate, questo errore è trascurabile, e FOMAML produce risultati empiricamente simili a MAML con la metà del costo computazionale.

La connessione tra FOMAML e Reptile (Nichol et al., 2018) è illuminante: Reptile esegue K passi di SGD interni e utilizza la differenza $(\Theta - \theta'_i)$ come direzione di aggiornamento, che è proporzionale alla media dei gradienti degli step interni. Si può dimostrare che, al secondo ordine in α, Reptile approssima MAML nella direzione di massimizzazione congiunta della performance finale dei task *e* del grado di sovrapposizione tra i gradienti dei task diversi — il che spiega perché Reptile tende a produrre inizializzazioni particolarmente adatte al transfer learning rapido.

Condizioni di convergenza

La convergenza di MAML al secondo ordine richiede che la funzione obiettivo esterna $\Sigma_i \mathcal{L}_{\mathcal{T}_i}(f_{\theta'_i(\Theta)})$ sia L-smooth rispetto a Θ, il che a sua volta richiede che le Hessiane delle loss dei singoli task siano limitate in norma spettrale. Sotto questa condizione, l'aggiornamento di SGD con learning rate η converge a un punto critico in $O(1/(\eta \, \varepsilon^2))$ passi nel regime non convesso. La velocità di adattamento interna α non influenza la velocità di convergenza esterna ma controlla il bias della soluzione: valori di α troppo grandi possono produrre oscillazioni nell'adattamento interno, valori troppo piccoli rendono insufficiente l'adattamento al task specifico.

B.4 MuZero: pianificazione MCTS e proprietà teoriche

Il framework del world model latente

MuZero separa nettamente la funzione di *rappresentazione* $h_\theta : \mathcal{O}^* \to \mathcal{S}$ (che mappa la storia delle osservazioni in uno stato latente), la funzione di *dinamica* $g_\theta : \mathcal{S} \times \mathcal{A} \to \mathcal{S} \times \mathbb{R}$ (che predice lo stato successivo e la ricompensa immediata), e la funzione di *predizione* $f_\theta : \mathcal{S} \to \Delta(\mathcal{A}) \times \mathbb{R}$ (che produce una prior sulle azioni e una stima del valore dallo stato corrente). A differenza di AlphaZero, MuZero non richiede accesso al modello simbolico dell'ambiente; a differenza di Dreamer, la pianificazione avviene con MCTS discreto piuttosto che con gradiente analitico su rollout continui.

UCB e la selezione nel MCTS

La regola di selezione UCB nell'albero di ricerca bilancia sfruttamento ed esplorazione. Per un nodo (s, a), il punteggio UCB è

$$\mathrm{UCB}(s, a) = Q(s, a) + c \cdot P(s, a) \cdot \sqrt{\textstyle\sum_b N(s, b)} \,/\, (1 + N(s, a)),$$

dove $Q(s, a) = R(s, a) + \gamma\, V(s')$ è la stima del valore azione, $N(s, a)$ è il contatore delle visite, $P(s, a)$ è la prior da f_θ, e $c > 0$ è un parametro di esplorazione. Questo punteggio è una variante del *PUCT* (Predictor + UCT) che incorpora la prior policy in modo inverso alla frequenza di visita. Si può dimostrare che, sotto condizioni di regolarità, con N simulazioni la regola UCB garantisce un ritorno di simulazione che converge al ritorno ottimale con velocità $O(\sqrt{\log N / N})$ — il classico bound di regret UCB1 adattato alla presenza di una prior.

Addestramento e target multipli

L'addestramento di MuZero è un problema di ottimizzazione multi-obiettivo: si vuole che g_θ predica ricompense accurate lungo K passi di unrolling nel modello, che f_θ predica valori accurati (via bootstrap da MCTS), e che la prior policy di f_θ imiti la distribuzione delle visite MCTS π (che è già una politica "migliorata" rispetto alla prior). La loss è

$$\mathcal{L} = \sum_t \sum_{k=0}^{K} \Big[(u_{t+k} - \hat{r}_{t+k}^{k})^2 + (z_{t+k} - \hat{v}_{t+k}^{k})^2 - \pi_{t+k}^{\mathsf{T}} \log \hat{p}_{t+k}^{k} \Big],$$

dove il superscripto k indica che il valore è ottenuto da k passi di unrolling del modello a partire dal tempo t, u_{t+k} è la ricompensa osservata, z_{t+k} è il target di valore bootstrappato, e π_{t+k}^{T} è la politica MCTS. Il gradiente non fluisce attraverso il modello di rappresentazione durante l'unrolling: g_θ riceve in input gli stati latenti prodotti da h_θ per il passo $k = 0$, e poi produce in cascata gli stati latenti per $k = 1, \ldots, K$.

Proprietà di consistenza del modello latente

Una questione teorica sottile riguarda la *consistenza* dello spazio latente: si ha garanzia che lo stato latente prodotto da g_θ dopo k passi di unrolling corrisponda allo stato latente che h_θ produrrebbe se applicato all'osservazione reale al passo t+k? In generale, no — MuZero non ha un vincolo esplicito di consistenza di questo tipo, e il modello latente può "deriva" rispetto all'osservazione reale nel corso dell'unrolling. Empiricamente, questo non sembra essere un problema grave per gli orizzonti di unrolling usati in pratica (K = 5), ma per orizzonti più lunghi può contribuire alla degradazione delle prestazioni su task con dinamiche complesse.

B.5 FlashAttention-2: analisi della complessità di I/O

Il collo di bottiglia della memoria nei Transformer

L'attenzione standard di Transformer ha complessità computazionale $O(n^2 d)$ e complessità di memoria $O(n^2)$ per sequenze di lunghezza n e dimensione del modello d, poiché la matrice di attenzione S ∈ ℝ^{n×n} deve essere materializzata in memoria per il calcolo del softmax e la ponderazione dei valori. Per n = 8192 (contesti lunghi tipici nei modelli del mondo su sequenze video), la matrice S richiederebbe 256MB in float32 — un fattore limitante critico per le GPU con memoria HBM di 40-80GB.

FlashAttention-2 riorganizza il calcolo dell'attenzione in modo da evitare la materializzazione della matrice completa S, accedendo alla memoria HBM in modo ottimale.

La chiave: online softmax e normalizzazione incrementale

Il softmax per la riga i della matrice di attenzione è $\text{softmax}(S_i)_j = \exp(S_{i□}) / \Sigma_k \exp(S_{i□})$. Il calcolo del denominatore richiede l'intera riga S_i, il che normalmente richiederebbe di caricare tutta la matrice K in memoria prima di poter calcolare il primo elemento dell'output. L'*online softmax* di Milakov e Gimelshein (2018) risolve questo problema con aggiornamenti incrementali: mantenendo il massimo corrente m e la somma parziale ℓ, ad ogni nuovo blocco di valori k si aggiornano

m_new = max(m_old, new_max), ℓ_new = exp(m_old − m_new) · ℓ_old + exp(new_block − m_new) · rowsum, O_new = diag(exp(m_old − m_new)) · O_old + exp(new_block − m_new) · P · V_block,

con la correzione finale O_final = diag(1/ℓ_new) · O_new. Questo schema permette di processare K e V in blocchi che restano in SRAM, senza materializzare mai la matrice S completa.

Analisi della complessità di I/O

Il modello HBM-SRAM di complessità di I/O (Aggarwal e Vitter, 1988) misura il numero di trasferimenti di dati tra memoria HBM (lenta, grande) e SRAM (veloce, piccola di dimensione M). L'attenzione standard richiede $\Theta(n^2)$ trasferimenti HBM per materializzare S e poi rileggerla per il backward pass. FlashAttention-2 riduce questo a $\Theta(n^2d/M)$ trasferimenti nel forward pass e $\Theta(n^2d/M)$ nel backward, poiché ogni blocco di S viene calcolato "on the fly" una sola volta (forward) o due volte (backward, dove è necessario ricalcolare S per non dover salvare l'intera matrice). La riduzione è di un fattore M/d rispetto all'attenzione standard — tipicamente un fattore 10-100 su hardware moderno.

Backward pass e la scelta di non salvare S

Il backward pass dell'attenzione standard propaga il gradiente attraverso $dO/dS = dO/dP \cdot P - \mathrm{diag}(P \cdot dO/dV^T \cdot V^T) \cdot P$, che richiede accesso a $P = \mathrm{softmax}(S)$. FlashAttention-2 ricalcola P on-the-fly durante il backward pass invece di salvarlo, impiegando i LOg-sum-exp (LSE) scalari $\ell_i = \log \Sigma \ \exp(S_i - m_i) + m_i$ che hanno già dimensione O(n) — trascurabile. Il trade-off tra ricomputazione e memoria è favorevole: la ricomputazione aggiunge un fattore costante al costo computazionale ma riduce drasticamente la pressione sulla memoria, rendendo possibili contesti molto più lunghi.

B.6 SAM: geometria del paesaggio di loss e proprietà di generalizzazione

Motivazione teorica: la connessione sharpness-generalizzazione

La *sharpness-aware minimization* (SAM) è motivata dalla congettura — supportata da evidenze empiriche e da alcuni risultati teorici parziali — che i minimi piatti del paesaggio di loss generalizzino meglio dei minimi acuti. Formalmente, si definisce la *sharpness* di un minimo θ^* come

$$S_\rho(\theta^*) = \max_{\|\varepsilon\|_2 \le \rho} \mathcal{L}(\theta^* + \varepsilon) - \mathcal{L}(\theta^*),$$

che misura il massimo deterioramento della loss in una sfera di raggio ρ. Il bound PAC-Bayesiano di Foret et al. (2021) mostra che, sotto una prior gaussiana e una distribuzione di test i.i.d., l'errore di generalizzazione atteso è limitato superiormente da

$$\mathcal{L}_{test} \le \mathcal{L}_{train} + O\left(\sqrt{\left(S_\rho(\theta^*) + \|\theta^*\|_2^2/\rho^2\right) / n}\right),$$

dove n è la dimensione del training set. Minimizzare la sharpness S_ρ ha dunque una giustificazione teorica diretta in termini di bound di generalizzazione.

L'algoritmo: un problema min-max approssimato

SAM risolve il problema di ottimizzazione

$\min_\theta \max_{\|\varepsilon\|_2 \le \rho} \mathcal{L}(\theta + \varepsilon)$.

Il massimo interno — trovare la perturbazione ε che massimizza la loss — è approssimato al primo ordine: linearizzando $\mathcal{L}(\theta + \varepsilon) \approx \mathcal{L}(\theta) + \varepsilon^T g$ dove $g = \nabla_\theta \mathcal{L}(\theta)$, il massimizzatore è

$\hat{\varepsilon} = \rho \cdot g / \|g\|_2$,

ovvero la perturbazione nella direzione del gradiente di norma esattamente ρ. Il gradiente di SAM rispetto ai parametri è quindi

$\nabla_\theta \mathcal{L}(\theta + \hat{\varepsilon}) \approx \nabla_\theta \mathcal{L}(\theta) \big|_{\theta = \theta + \hat{\varepsilon}}$,

calcolato nel punto perturbato $\theta + \hat{\varepsilon}$. Questo richiede due forward e due backward pass per iterazione — un raddoppio del costo computazionale — ma può essere implementato senza modifiche all'architettura.

ASAM e l'invarianza di scala

Una debolezza di SAM è la dipendenza dalla scala dei parametri: se i parametri di un layer vengono scalati di un fattore λ, la sharpness S_ρ cambia ma l'ipersuperficie geometrica rappresentata dalla rete non cambia (per funzioni positive-omogenee come ReLU). L'*Adaptive SAM* (ASAM) corregge questo sostituendo la palla euclidea con una palla nella metrica adattiva:

$\hat{\varepsilon}_{ASAM} = \rho \cdot |\theta| \odot g / \||\theta| \odot g\|_2$,

dove $|\theta|$ denota il vettore dei valori assoluti dei parametri. Questa scelta rende l'algoritmo invariante rispetto a riscalamenti dei parametri di singoli layer, una proprietà desiderabile per reti profonde con BatchNorm o LayerNorm.

B.7 MPPI: ottimizzazione stocastica per il controllo ottimale

Derivazione dal principio del cammino integrale

Il controllo ottimale stocastico cerca la sequenza di azioni $U^* = (u_0^*, ..., u_{H-1}^*)$ che minimizza il costo atteso

$J(U) = \mathbb{E}_{\tau \sim p(\tau|U)} [\sum_{t=0}^{H} c(s_t, u_t)]$,

dove $p(\tau|U)$ è la distribuzione sulle traiettorie indotta dalla politica di controllo U e dalla dinamica stocastica f. Il *Model Predictive Path Integral* (MPPI) deriva la soluzione ottimale a partire dalla teoria del controllo stocastico di Williams et al. (2017) basandosi sulla connessione tra distribuzione di cammini ottimale e importanza campionaria.

Dati N campioni di traiettorie $\tau^{(k)}$ generate dalla distribuzione di campionamento $q(\tau) = \mathcal{N}(U, \sigma^2 I)^H$, e i relativi costi $S(\tau^{(k)}) = \Sigma_t\, c(s_t^{(k)}, u_t^{(k)})$, il controllo ottimale approssimato è

$$U^* \approx \Sigma_k\, w_k\, U^{(k)} \,/\, \Sigma_j\, w_j,\ w_k = \exp(-S(\tau^{(k)}) / \lambda),$$

dove $\lambda > 0$ è il parametro di temperatura che controlla la selettività della media pesata. Questa formula è un'approssimazione della soluzione esatta del problema di controllo stocastico: si può dimostrare che, all'aumentare di N, la stima converge alla soluzione del problema

$$\min_U KL[q(\tau) \,\|\, p^*(\tau)] \text{ s.t. } \mathbb{E}_q[S(\tau)] = \text{costo target},$$

dove $p^*(\tau) \propto \exp(-S(\tau)/\lambda)\ q(\tau)$ è la distribuzione di cammini ottimale nella formulazione di Williams-Theodorou.

Proprietà di convergenza e campionamento

La qualità dell'approssimazione MPPI dipende dalla *effective sample size* (ESS)

$$ESS = (\Sigma_k\, w_k)^2 \,/\, \Sigma_k\, w_k^2,$$

che misura il numero effettivo di campioni non ridondanti. Quando i costi $S(\tau^{(k)})$ sono molto concentrati (distribuzione di costo molto peakada), pochi campioni dominano la media pesata e $ESS \ll N$, degradando la qualità dell'ottimizzazione. La scelta del parametro di temperatura λ controlla direttamente questo fenomeno: valori grandi di λ rendono la media pesata più uniforme (alta ESS, bassa selettività), valori piccoli la rendono più selettiva (bassa ESS, alta precisione). In pratica si usa λ adattivo o si normalizzano i costi con $\beta_{min} = \min_k S(\tau^{(k)})$ come sottraendo stabile: $w_k = \exp(-(S(\tau^{(k)}) - \beta_{min})/\lambda)$, che elimina il rischio di overflow numerico senza cambiare le proporzioni dei pesi.

Complessità e parallelizzazione

MPPI è intrinsecamente parallelizzabile: i N rollout possono essere eseguiti simultaneamente su GPU, con costo $O(N \cdot H \cdot d_s)$ per il rollout del modello e $O(N)$ per il calcolo dei pesi. Il collo di bottiglia computazionale nella pratica è il modello f: se f è una rete neurale profonda, il costo dei N rollout è N volte il costo di un singolo forward pass, che può essere significativo per modelli di dimensione medio-grande. Tecniche di parallelizzazione su batch (con vectorized env o su più GPU) permettono di portare questo costo a $O(H \cdot d_s)$ in tempo con N processori, rendendo MPPI

praticabile in tempo reale per sistemi con orizzonti corti ($H \leq 50$) e modelli di transizione leggeri.

Appendice C — Valutazione Empirica: Benchmark, Dataset e Metodologia Statistica

---

Premessa

La valutazione empirica dei modelli del mondo è un problema statistico di considerevole complessità, non riducibile alla pubblicazione di tabelle di risultati. Questa appendice discute i benchmark e i dataset più rilevanti per la monografia, ma lo fa affrontando preliminarmente la questione metodologica: che cosa misura un benchmark, come si interpretano i risultati in modo statisticamente corretto, e quali sono le fonti di variabilità e di bias sistematico che rendono i confronti tra architetture diverse più incerti di quanto la letteratura tenda ad ammettere. Solo dopo aver stabilito questo quadro metodologico si descrivono i singoli ambienti e dataset, collocandoli nella struttura argomentativa dell'opera.

C.1 Il problema statistico della valutazione dei world model

La valutazione di un sistema di apprendimento per rinforzo model-based richiede di stimare la performance attesa su una distribuzione di compiti, partendo da un numero limitato di esecuzioni (*seeds*) con condizioni iniziali diverse. Il rendimento di un agente su un compito specifico è una variabile aleatoria — dipende dall'inizializzazione dei parametri, dalla sequenza di dati di addestramento, e in alcuni ambienti da elementi stocastici della dinamica — e la sua distribuzione è tipicamente asimmetrica, multimodale e con code pesanti, il che rende la media aritmetica uno stimatore di posizione non robusto.

Il problema diventa più acuto nei confronti tra architetture diverse: si vuole rispondere alla domanda "l'architettura A è migliore dell'architettura B sul benchmark X", che è formalmente un test di ipotesi. Il test standard nella letteratura — riportare medie e deviazioni standard su 3-5 seed — è spesso insufficiente. Studi di replicabilità (Agarwal et al., 2021) hanno mostrato che la variabilità tra seed può essere così elevata da rendere i confronti pubblicati statisticamente non significativi anche quando la differenza nelle medie appare sostanziale.

La metodologia statistica adeguata per i benchmark di RL comprende: l'uso di un numero sufficiente di seed (tipicamente ≥ 10), la stima degli intervalli di confidenza su metriche aggregate tramite bootstrap, la normalizzazione dei punteggi rispetto a un baseline comune per permettere confronti tra compiti con scale diverse, e il test di Mann-Whitney U come alternativa non parametrica al t-test per confronti tra distribuzioni asimmetriche. Il *Interquartile Mean* (IQM) — media sul quartile centrale della distribuzione dei punteggi — è più robusto alle code pesanti rispetto alla media aritmetica pur mantenendo interpretabilità.

La Performance Profile (PP) di Agarwal et al. è uno strumento grafico che, per un valore di soglia τ, mostra la frazione di esecuzioni in cui un algoritmo raggiunge almeno una performance normalizzata τ; dominanza uniforme nella PP implica dominanza stocastica di primo ordine, il che costituisce un'affermazione più forte della semplice superiorità della media.

C.2 DeepMind Control Suite

Il *DeepMind Control Suite* (DMControl, Tassa et al., 2018) è l'ambiente di riferimento per il controllo model-based a bassa dimensionalità. Comprende 28 compiti di controllo motorio continuo su corpi fisici articolati — Walker (camminata e corsa), Cheetah (corsa veloce), Hopper (saltello su un piede), Cartpole (bilanciamento), Reacher (raggiungimento di un target), Acrobot (manipolazione), Humanoid (locomozione bipede) — con varianti che differiscono per difficoltà (easy/hard) e ricchezza osservazionale (stati propriocettivi vs. pixel della telecamera).

Le metriche standard sono il *punteggio episodico* (episode return) normalizzato tra 0 e 1000, misurato dopo un numero fisso di step di interazione con l'ambiente (tipicamente 100k, 500k o 1M step). Il confronto a 500k step su pixel è il regime più impegnativo: gli algoritmi model-free come SAC richiedono tipicamente 1M-3M step per raggiungere prestazioni comparabili a quelle dei modelli model-based come DreamerV3 o TD-MPC2 a 500k step, il che giustifica empiricamente il vantaggio di efficienza campionaria attribuito ai world model.

Le difficoltà metodologiche specifiche di DMControl riguardano la sensibilità all'inizializzazione del seed fisico (le configurazioni fisiche di partenza degli agenti), la non stazionarietà della curva di apprendimento (alcuni agenti hanno progressi rapidi iniziali seguiti da plateau prolungati), e la differenza tra prestazioni "final" (punteggio all'ultimo episodio) e prestazioni "area sotto la curva" (integrale del punteggio nel corso dell'addestramento), che danno indicazioni diverse sull'efficienza di apprendimento.

C.3 Atari 100k

Il benchmark *Atari 100k* (Kaiser et al., 2020) misura la performance su 26 giochi Atari dopo soli 100k step di interazione — equivalenti a circa 2 ore di gioco umano — in netto contrasto con i 200M step tradizionalmente usati nei benchmark Atari completi. La drastica riduzione del budget di interazione sposta il focus dall'asintoto di performance all'efficienza campionaria: un sistema che apprende rapidamente da poche esperienze, come ci si aspetterebbe da un agente con un buon modello del mondo.

La normalizzazione dei punteggi su Atari 100k segue la convenzione di Kaiser et al.: per ogni gioco g, il punteggio normalizzato è (score_algorithm − score_random) / (score_human − score_random), così che 0 corrisponde alla performance di un agente casuale e 1 alla performance umana. Il *Human Normalized Score* (HNS) medio su tutti i 26 giochi è la metrica aggregata standard. DreamerV3 raggiunge un HNS medio di circa 1.14 su Atari 100k (superando la performance umana media), mentre gli algoritmi model-free come Data Efficient Rainbow si fermano tipicamente intorno a 0.35-0.40 con lo stesso budget.

Una criticità metodologica del benchmark Atari 100k è la varianza estremamente elevata tra seed su alcuni giochi specifici (in particolare su Pitfall, Montezuma's Revenge e altri giochi con ricompensa sparsa), che possono dominare la variabilità dell'IQM aggregato. Una valutazione rigorosa dovrebbe riportare i risultati per-gioco oltre all'aggregato, e impiegare la Performance Profile per caratterizzare la distribuzione dei punteggi.

C.4 D4RL: benchmark per il reinforcement learning offline

Il benchmark *D4RL* (Fu et al., 2020) è stato progettato specificamente per la valutazione del reinforcement learning offline — il regime in cui l'agente deve apprendere una politica da un dataset fisso di traiettorie senza interazione con l'ambiente. La sua rilevanza per la monografia è diretta: è il benchmark su cui vengono valutati gli algoritmi MOPO, MOReL, MOBILE e le loro varianti discusse nel capitolo sull'offline MBRL.

D4RL comprende dataset di quattro tipi per ciascun compito motorio di DMControl: *random* (traiettorie generate da una politica casuale, bassa copertura), *medium* (traiettorie da una politica parzialmente addestrata, copertura intermedia), *medium-replay* (il replay buffer di una politica in addestramento, copertura variabile), e *medium-expert* (mix di traiettorie di media e alta qualità, alta copertura nelle regioni di alta performance). Questa stratificazione permette di valutare la robustezza degli algoritmi a diverse qualità e distribuzioni dei dati.

La metrica normalizzata di D4RL è analoga a quella di Atari: (score_algorithm − score_random) / (score_expert − score_random), con 100 che corrisponde alla

performance di una politica esperta. Gli algoritmi model-based offline (MOPO, MOBILE) tipicamente superano i baseline model-free (BC, IQL) sui dataset di tipo medium e medium-replay, dove il modello del mondo può effettivamente contribuire all'esplorazione conservativa, ma non necessariamente sui dataset random dove la scarsa copertura rende la simulazione immaginaria altamente inaffidabile.

Un aspetto metodologico critico specifico di D4RL: la scelta dell'iperparametro di penalizzazione dell'incertezza (il parametro $\varkappa$ in MOPO o λ in MOBILE) è fortemente dipendente dal dataset, e la selezione di questo iperparametro richiede formalmente un insieme di validazione separato dal training set offline. Molti risultati pubblicati ottimizzano questo parametro sull'insieme di test, il che introduce un bias ottimistico nelle stime di performance — una violazione metodologica che non sempre è segnalata esplicitamente.

C.5 Dataset per l'apprendimento delle rappresentazioni visive

La valutazione di I-JEPA e delle architetture JEPA visive si svolge su un insieme di benchmark distinti da quelli del reinforcement learning, orientati alla qualità delle rappresentazioni apprese piuttosto che alla performance di controllo.

ImageNet-1k (Deng et al., 2009) nella variante *linear probing* è il benchmark standard per la valutazione delle rappresentazioni visive auto-supervisionate: l'encoder viene preaddestrato su ImageNet senza etichette, e poi si addestra un classificatore lineare *top-frozen* sull'encoder. La performance di classificazione su 1000 classi misura quanto le rappresentazioni apprese contengano informazione semanticamente discriminativa in forma linearmente separabile. I-JEPA (Assran et al., 2023) raggiunge un'accuratezza top-1 di 76.5% con ViT-H su ImageNet-1k senza fine-tuning completo, comparabile o superiore a SimCLR, BYOL e DINO con risorse computazionali di preaddestramento comparabili.

La valutazione *few-shot* su benchmark come ImageNet-5-shot o i benchmarks di Gidaris et al. misura la capacità delle rappresentazioni di generalizzare a nuove categorie con pochissimi esempi etichettati, che è più direttamente rilevante per l'uso delle rappresentazioni in sistemi di pianificazione e controllo robotico.

La valutazione della qualità delle rappresentazioni temporali di V-JEPA avviene su benchmark video come Kinetics-400 (riconoscimento di azioni) e Something-Something v2 (comprensione di relazioni causali fisiche), dove la struttura temporale dell'input è essenziale per la classificazione corretta. La performance su Something-Something v2 è considerata una misura più diretta della comprensione del movimento fisico rispetto a Kinetics, che è più dominato da caratteristiche spaziali statiche.

C.6 Benchmark per la comprensione fisica e causale

La valutazione della comprensione causale e fisica dei modelli del mondo va al di là dei benchmark di classificazione o controllo tradizionali, e richiede test progettati specificamente per sondare la struttura del modello interno appreso. Questo insieme di benchmark è meno consolidato rispetto a DMControl o ImageNet, ma è di importanza crescente per la tesi argomentata in questa monografia.

PHYRE (Bakhtin et al., 2019) valuta la capacità di predire le conseguenze fisiche di azioni in ambienti di simulazione 2D, richiedendo di identificare quale azione iniziale porta a un determinato risultato finale. Un sistema con un buon modello del mondo fisico dovrebbe eccellere su PHYRE; i risultati attuali mostrano che i modelli neurali che operano direttamente su pixel hanno prestazioni inferiori a quelle di sistemi che incorporano simulatori fisici espliciti, suggerendo che i world model impliciti attuali non internalizzano completamente la fisica newtoniana.

CausalWorld (Ahmed et al., 2020) è un benchmark robotico che richiede la manipolazione di oggetti fisici con struttura causale esplicita — costruire torri, spostare oggetti in posizioni target — con una sistematica di test fuori distribuzione che separa l'interpolazione dall'estrapolazione causale. La performance fuori distribuzione è significativamente più bassa di quella in distribuzione per tutti i modelli del mondo neurali attuali, una lacuna che il capitolo sull'apprendimento causale di questa monografia analizza in profondità.

BabyAI (Chevalier-Boisvert et al., 2019) è un ambiente linguistico-visivo in cui un agente deve seguire istruzioni in linguaggio naturale in ambienti di griglia 2D. La sua rilevanza per i modelli del mondo multimodali — discussi nel capitolo sul grounding visio-linguistico — è che richiede un'integrazione tra comprensione del linguaggio e pianificazione spaziale che nessuno dei due moduli da solo è in grado di gestire.

C.7 Ambienti per il sim-to-real transfer

La valutazione del trasferimento sim-to-real è intrinsecamente più complessa di quella in ambienti puramente simulati: l'ambiente reale introduce variabilità non controllabile (friction variabile, inerzia degli attuatori, ritardi di comunicazione, occlusioni impreviste), e la raccolta di dati reali è costosa e lenta rispetto alla simulazione. La strategia di valutazione standard è la *domain randomization sistematica*: si addestra il modello del mondo su una distribuzione di parametri fisici simulati, e si valuta il trasferimento su intervalli di variazione controllata. La *sim-to-real gap* è la differenza di performance tra il sistema nel miglior simulatore disponibile e nel sistema fisico reale, con parametri fisici nominali.

Il dataset *Open X-Embodiment* (Open X-Embodiment Collaboration, 2023) aggrega oltre 1 milione di traiettorie di manipolazione robotica da 22 tipi di robot diversi, raccolte da numerosi laboratori. È il dataset di riferimento per la valutazione dei modelli del mondo robotici *fondazionali* — preaddestrati su dati eterogenei e poi adattati a compiti specifici — e pone sfide metodologiche significative: la distribuzione dei task è altamente sbilanciata, le configurazioni di robot e ambienti sono eterogenee, e la metrica di successo varia da task a task. La normalizzazione dei punteggi attraverso task e robot diversi rimane un problema aperto, e i risultati pubblicati su Open X-Embodiment devono essere interpretati con cautela riguardo al significato dell'aggregazione.

C.8 Linee guida per la riproducibilità

L'appendice si chiude con alcune raccomandazioni metodologiche per la valutazione rigorosa dei modelli del mondo, derivate dalla letteratura sulla riproducibilità in apprendimento automatico (Pineau et al., 2021; Agarwal et al., 2021).

L'algoritmo dovrebbe essere valutato su almeno 10 seed indipendenti per ciascun compito; i risultati dovrebbero riportare la distribuzione completa dei punteggi (non solo media e deviazione standard) attraverso il Performance Profile; la selezione degli iperparametri dovrebbe avvenire su un insieme di compiti distinto da quello di valutazione finale, e il processo di selezione dovrebbe essere descritto esplicitamente; il codice sorgente e i checkpoint del modello dovrebbero essere pubblicamente disponibili per permettere la verifica indipendente.

Quando la comparazione tra algoritmi è lo scopo primario, il test di ipotesi appropriato dipende dalla distribuzione dei punteggi: per distribuzioni approssimativamente normali si usa il t-test appaiato; per distribuzioni asimmetriche o multimodali si preferisce il test di Wilcoxon signed-rank; per confronti multipli simultanei si applica la correzione di Bonferroni o Holm per il controllo del family-wise error rate. In tutti i casi, la significatività statistica è condizione necessaria ma non sufficiente per l'affidabilità scientifica di un confronto: occorre che la differenza osservata sia anche *praticamente significativa*, ovvero di entità sufficiente da avere rilevanza per le applicazioni di interesse.

Appendice D — Librerie Software e Framework Implementativi

La ricerca sui world model si appoggia su un ecosistema software in rapida evoluzione. Questa appendice cataloga i framework principali con le versioni di riferimento alla data di stesura del testo (dicembre 2024).

D.1 Framework di Deep Learning

Libreria	Versione ref.	Caratteristiche principali per i world model
PyTorch	≥ 2.2	Autograd dinamico; torch.compile; FlashAttention integrata; torch.func per MAML
JAX + Flax	JAX 0.4+	XLA compilazione; jit, vmap, grad compostabili; ideale per Neural ODE e Hamiltonian NN
TensorFlow/Keras	2.15+	tf.keras; TF-Agents per RL; TPU nativo
Equinox (JAX)	0.11+	OOP pytree-based; ideale per architetture custom di world model

D.2 Librerie per Reinforcement Learning e World Model

Libreria / Repository	Descrizione
DreamerV3 (danijar/dreamer)	Implementazione ufficiale JAX di DreamerV3; supporta task continui e discreti
TDMPC2 (nicklashansen)	TD-MPC2 PyTorch; planner model-based efficiente per DMControl e Meta-World
stable-baselines3	Implementazioni affidabili PPO, SAC, TQC; utilizzate come baseline model-free
RLlib (Ray)	Framework distribuito; multi-agent; callback per world model custom
mbrl-lib (Facebook)	Libreria per MBRL con PETS, MBPO, Dreamer; modello-standard per ablation study
Acme (DeepMind)	Distributed RL; MuZero, AlphaZero ufficiali DeepMind
tianshou	Framework PyTorch modulare; ottimo per prototipazione rapida di algoritmi custom

D.3 SSL e Rappresentazioni Latenti

Libreria	Descrizione
JEPA (facebookresearch)	Implementazione ufficiale I-JEPA, V-JEPA 1/2 di Meta AI in PyTorch
timm (rwightman)	Raccolta di ViT pretrainati; supporto fine-tuning I-JEPA, DINO, MAE
vissl (Facebook)	Framework SSL: SimCLR, BYOL, SwAV, Barlow Twins, VICReg
lightly (lightly-ai)	Library SSL leggera; easy integration con PyTorch Lightning
DINOv2 (facebookresearch)	Implementazione ufficiale DINOv2; ViT-G/14 features; DPT probing heads
open_clip	CLIP e varianti (SigLIP, ALIGN) open-source; pesi pre-trainati su LAION

D.4 Modelli Generativi e Diffusione

Libreria	Descrizione
diffusers (HuggingFace)	Libreria standard DDPM/DDIM/Flow Matching/LDM; backbone UniSim, π_0
torchdiffeq	ODE/SDE solvers differenziabili; Neural ODE, Hamiltonian NN, Latent ODE
torchsde	SDE stocastiche differenziabili; Neural SDE, Score-based models
zuko	Normalizing flows in PyTorch; coupling layers, MAF, NSF
normflows	Altro framework NF PyTorch; CNF con torchdiffeq
POT (Python OT)	Trasporto ottimale; Sinkhorn; Wasserstein distances; Schrödinger bridge

D.5 Geometria Differenziale e Topologia

Libreria	Descrizione
geomstats	Geometria riemanniana: SO(3), SE(3), Lie groups; statistiche su manifold
geoopt	Ottimizzazione riemanniana su PyTorch; RiemannianAdam; spazi iperbolici
giotto-tda	Topological Data Analysis; persistent homology su Python (Ripser backend)
Ripser++	GPU-accelerated persistent homology; calcolo rapido per cloud di punti grandi
gudhi	Libreria C++/Python completa: Čech, Rips, Mapper, Witness complexes
persim	Distanze tra diagrammi di persistenza (bottleneck, Wasserstein)

D.6 Causal Inference e Structural Learning

Libreria	Descrizione
DoWhy	Framework causale Microsoft; do-calculus, refutation tests
CausalML (Uber)	Causal inference per recommendation; uplift modeling
NOTEARS (bryanllg)	Implementazione ufficiale NOTEARS e NOTEARS-MLP per struttura DAG

cdt (Diviyan)	Causal Discovery Toolbox; PC, FCI, CGNN
tigramite	PCMCI per time-series causal discovery; usato per analisi world model latente

Appendice E — Tabelle Comparative tra Architetture

Questa appendice sintetizza le caratteristiche architetturali, computazionali e di performance delle principali architetture di world model discusse nel testo, con l'obiettivo di facilitare la selezione del paradigma appropriato per diversi contesti applicativi.

E.1 Confronto tra World Model Profondi per RL

Modello	Backbone	Spazio stati	Azione	Costo comp.	Punto di forza / limite principale
PlaNet	GRU + RSSM	Gauss. cont.	CEM	Basso	Primo RSSM; solo task bassa dim. azione
DreamerV1	GRU + RSSM	Gauss. cont.	Actor-Critic	Medio	Introduce AC imaginario; nessun tuning β
DreamerV2	GRU + RSSM	Categ. K×C	Actor-Critic	Medio	Categoriali per stabilità; Atari SOTA
DreamerV3	GRU + RSSM	Categ. K×C	Actor-Critic	Medio-alto	KL_sym + reward norm; robusto cross-domain
IRIS	VQVAE + Transf	Discreto	Transf. AR	Alto	Atari 100K SOTA; costoso per spazi continui
TWM	RSSM + Transf	Ibrido	Actor-Critic	Alto	Dipendenze lungo termine; scalabile
TD-MPC2	MLP + TOLD	Latente	MPPI	Basso	Efficienza computazionale; ottimo per robotica
MuZero	ResNet	Latente	MCTS	Molto alto	Board games SOTA; oneroso per ambienti continui

E.2 Confronto tra Metodi SSL Visivi

Metodo	Tipo loss	Negativi	Augmentazioni	Collapse?	Caratteristica chiave
SimCLR	InfoNCE	In-batch	Pesanti	No (negativi)	Semplice; richiede grandi batch
MoCo v3	InfoNCE	Queue + EMA	Pesanti	No (queue)	Memory bank; batch più piccoli
BYOL	MSE coseno	No negativi	Pesanti	No (EMA)	Senza negativi; EMA anti-collasso

SimSiam	MSE coseno	No negativi	Pesanti	No (stop-grad)	Stop-gradient; teoria EM implicita
VICReg	Inv+Var+Cov	No negativi	Moderate	No (Var reg)	Whitening implicito; interpretabile
Barlow Twins	Cross-corr	No negativi	Moderate	No (off-diag)	Riduzione ridondanza decorrelando canali
DINO v2	Cross-entropy	No negativi	Moderate	No (centering)	Segmentazione emerge; 142M img
MAE	MSE pixel	N/A	Nessuna	N/A	75% masking; decoder leggero; alta efficienza
I-JEPA	MSE latente	No negativi	Nessuna	No (EMA)	Predizione latente; migliore semantica vs MAE
V-JEPA 2	MSE latente	No negativi	Nessuna	No (EMA)	3D spatio-temp.; comprensione fisica video

E.3 Prestazioni su DMControl (500k steps, 1 seed medio)

Nota: i valori sono indicativi delle medie riportate in letteratura; la varianza tra seed può essere significativa. Valori in termini di score normalizzato 0–1000 (cifre arrotondate).

Algoritmo	Cheetah	Walker	Reacher	Humanoid	Sample eff.
DreamerV2	800	920	950	350	★★★★☆
DreamerV3	870	960	970	620	★★★★☆
TD-MPC2	890	970	980	730	★★★★★
SAC (model-free)	820	850	920	280	★★★☆☆
PETS (ensemble)	630	780	880	—	★★★☆☆

E.4 Confronto tra Architetture di Memoria per World Model

Architettura	Capacità	Accesso	Aggiorn. online	Caso d'uso tipico
GRU / LSTM	O(d)	O(1)	Sì (gradiente)	History compatta; world model sequenziale
Transformer (KV-cache)	O(n·d)	$O(n^2)$	Parziale (append)	Ragionamento long-context; RSSM + attention
NTM / DNC	O(N·d)	O(N)	Sì (diff.)	Task di copia/recupero strutturato; maze nav.
kNN episodica (NEC)	O(|buffer|)	O(log N)	One-shot	Adattamento rapido; no catastrophic forgetting

NeRF implicita	O(|θ|)	O(ray)	SGD incrementale	Mappa 3D implicita; SLAM neurale
Transformer-XL	O(L·M·d)	O(L·M)	Parziale (cache)	World model a segmenti lunghi
Compressive Transf.	O((M+M_c)d)	O(M+M_c)	Compressione	Long-range video world model

Bibliografia

La bibliografia è organizzata in sezioni tematiche corrispondenti ai capitoli principali della monografia, con ordinamento alfabetico per autore all'interno di ogni sezione. Per i preprint arXiv, si riporta l'anno della prima pubblicazione; per i paper con versioni riviste, l'anno della versione più citata nel testo.

Fondamenti e Modelli del Mondo Classici

[Atari-DQN] Mnih, V. et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540), 529–533.

[Baddeley-1974] Baddeley, A. D., & Hitch, G. J. (1974). Working memory. *Psychology of Learning and Motivation*, 8, 47–89.

[Craik-1943] Craik, K. J. W. (1943). *The Nature of Explanation*. Cambridge University Press.

[Friston-2010] Friston, K. (2010). The free-energy principle: a unified brain theory? *Nature Reviews Neuroscience*, 11(2), 127–138.

[Gibson-1979] Gibson, J. J. (1979). *The Ecological Approach to Visual Perception*. Houghton Mifflin.

[Ha-Schmidhuber-2018] Ha, D., & Schmidhuber, J. (2018). World models. *arXiv:1803.10122*.

[Kalman-1960] Kalman, R. E. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, 82(1), 35–45.

[LeCun-2022] LeCun, Y. (2022). A path towards autonomous machine intelligence. *OpenReview*, version 0.9.2.

[Pearl-2000] Pearl, J. (2000). *Causality: Models, Reasoning, and Inference*. Cambridge University Press.

[Sutton-1990] Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. *Machine Learning Proceedings*, pp. 216–224.

[Tolman-1948] Tolman, E. C. (1948). Cognitive maps in rats and men. *Psychological Review*, 55(4), 189–208.

[Tishby-2000] Tishby, N., Pereira, F., & Bialek, W. (2000). The information bottleneck method. *arXiv:physics/0004057*.

[Wiener-1948] Wiener, N. (1948). *Cybernetics: Or Control and Communication in the Animal and the Machine*. MIT Press.

Variational Autoencoders, RSSM e Dreamer

[DreamerV1] Hafner, D., Lillicrap, T., Ba, J., & Norouzi, M. (2020). Dream to control: Learning behaviors by latent imagination. *ICLR 2020*.

[DreamerV2] Hafner, D., Lillicrap, T., Norouzi, M., & Ba, J. (2021). Mastering Atari with discrete world models. *ICLR 2021*.

[DreamerV3] Hafner, D., Hafner, J., Pasukonis, J., Lillicrap, T., & Norouzi, M. (2023). Mastering diverse domains through world models. *arXiv:2301.04104*.

[IRIS] Micheli, V., Alonso, E., & Fleuret, F. (2022). Transformers are sample-efficient world models. *ICLR 2023*.

[Kingma-Welling-2014] Kingma, D. P., & Welling, M. (2014). Auto-encoding variational Bayes. *ICLR 2014*.

[PlaNet] Hafner, D. et al. (2019). Learning latent dynamics for planning from pixels. *ICML 2019*.

[RSSM-Hafner] Hafner, D. et al. (2019). Recurrent world models facilitate policy evolution. *NeurIPS 2019*.

[Rezende-2014] Rezende, D., Mohamed, S., & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. *ICML 2014*.

[TWM] Robine, J., Höftmann, M., Uelwer, T., & Harmeling, S. (2023). Transformer-based world models are happy with 100k interactions. *ICLR 2023*.

JEPA e Apprendimento Self-Supervised

[Assran-IJEPA] Assran, M. et al. (2023). Self-supervised learning from images with a joint-embedding predictive architecture. *CVPR 2023*.

[Bardes-VJEPA] Bardes, A. et al. (2024). V-JEPA: Latent video prediction for visual representation learning. *arXiv:2404.08471*.

[Barlow-Twins] Zbontar, J. et al. (2021). Barlow Twins: Self-supervised learning via redundancy reduction. *ICML 2021*.

[BYOL] Grill, J.-B. et al. (2020). Bootstrap your own latent: A new approach to self-supervised learning. *NeurIPS 2020*.

[CPC] van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *arXiv:1807.03748*.

[DINO] Caron, M. et al. (2021). Emerging properties in self-supervised vision transformers. *ICCV 2021*.

[DINOv2] Oquab, M. et al. (2023). DINOv2: Learning robust visual features without supervision. *TMLR 2024*.

[MAE] He, K. et al. (2022). Masked autoencoders are scalable vision learners. *CVPR 2022*.

[MoCo-v3] Chen, X., & He, K. (2021). Exploring simple Siamese representation learning. *CVPR 2021*.

[SimCLR] Chen, T., Kornblith, S., Norouzi, M., & Hinton, G. (2020). A simple framework for contrastive learning. *ICML 2020*.

[VICReg] Bardes, A., Ponce, J., & LeCun, Y. (2022). VICReg: Variance-invariance-covariance regularization for self-supervised learning. *ICLR 2022*.

Architetture Transformer e Attenzione

[Attention-Is-All-You-Need] Vaswani, A. et al. (2017). Attention is all you need. *NeurIPS 2017*.

[BEiT] Bao, H. et al. (2022). BEiT: BERT pre-training of image transformers. *ICLR 2022*.

[Flash-Attention] Dao, T. et al. (2022). FlashAttention: Fast and memory-efficient exact attention. *NeurIPS 2022*.

[Flash-Attention-2] Dao, T. (2023). FlashAttention-2: Faster attention with better parallelism. *ICLR 2024*.

[Flash-Attention-3] Shah, J. et al. (2024). FlashAttention-3: Fast and accurate attention on H100 GPUs. *arXiv:2407.08608*.

[Genie] Bruce, J. et al. (2024). Genie: Generative interactive environments. *ICML 2024*.

[Mamba] Gu, A., & Dao, T. (2023). Mamba: Linear-time sequence modeling with selective state spaces. *arXiv:2312.00752*.

[Modern-Hopfield] Ramsauer, H. et al. (2021). Hopfield networks is all you need. *ICLR 2021*.

[RoPE] Su, J. et al. (2021). RoFormer: Enhanced transformer with rotary position embedding. *arXiv:2104.09864*.

[Transformer-XL] Dai, Z. et al. (2019). Transformer-XL: Attentive language models beyond a fixed-length context. *ACL 2019*.

[ViT] Dosovitskiy, A. et al. (2021). An image is worth 16×16 words. *ICLR 2021*.

Pianificazione e Reinforcement Learning Model-Based

[AlphaGo] Silver, D. et al. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature*, 529, 484–489.

[AlphaZero] Silver, D. et al. (2018). A general reinforcement learning algorithm that masters chess, shogi, and Go. *Science*, 362(6419), 1140–1144.

[CEM] Rubinstein, R. Y., & Kroese, D. P. (2004). *The Cross-Entropy Method*. Springer.

[DAgger] Ross, S., Gordon, G., & Bagnell, D. (2011). A reduction of imitation learning to no-regret online learning. *AISTATS 2011*.

[Diffusion-Policy] Chi, C. et al. (2023). Diffusion policy: Visuomotor policy learning via action diffusion. *RSS 2023*.

[EfficientZero] Ye, W. et al. (2021). Mastering Atari games with limited data. *NeurIPS 2021*.

[MPPI] Williams, G. et al. (2017). Information-theoretic MPC for model-based reinforcement learning. *ICRA 2017*.

[MuZero] Schrittwieser, J. et al. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588, 604–609.

[PPO] Schulman, J. et al. (2017). Proximal policy optimization algorithms. *arXiv:1707.06347*.

[SAC] Haarnoja, T. et al. (2018). Soft actor-critic. *ICML 2018*.

[Sampled-MuZero] Hubert, T. et al. (2021). Learning and planning in complex action spaces. *ICML 2021*.

Robotica Incarnata e Controllo Model-Based

[Andrychowicz-2020] Andrychowicz, M. et al. (2020). Learning dexterous in-hand manipulation. *IJRR*, 39(1), 3–20.

[Brooks-1991] Brooks, R. A. (1991). Intelligence without representation. *Artificial Intelligence*, 47(1–3), 139–159.

[Domain-Rand] Tobin, J. et al. (2017). Domain randomization for transferring deep neural networks from simulation to the real world. *IROS 2017*.

[Gato] Reed, S. et al. (2022). A generalist agent. *TMLR 2022*.

[OpenVLA] Kim, M. J. et al. (2024). OpenVLA: An open-source vision-language-action model. *arXiv:2406.09246*.

[Pi-Zero] Black, K. et al. (2024). π_0: A vision-language-action flow model for general robot control. *arXiv:2410.24164*.

[RT-2] Brohan, A. et al. (2023). RT-2: Vision-language-action models transfer web knowledge to robotic control. *CoRL 2023*.

[RT-X] Open X-Embodiment Collaboration (2023). Open X-Embodiment: Robotic learning datasets. *arXiv:2310.08864*.

[SayCan] Ahn, M. et al. (2022). Do as I can, not as I say: Grounding language in robotic affordances. *CoRL 2022*.

[Varela-1991] Varela, F. J., Thompson, E., & Rosch, E. (1991). *The Embodied Mind*. MIT Press.

Neural ODE, Reti Hamiltoniane e Dinamica Continua

[Chen-NeuralODE] Chen, R. T. Q. et al. (2018). Neural ordinary differential equations. *NeurIPS 2018*.

[EGNN] Satorras, V. G. et al. (2021). E(n) equivariant graph neural networks. *ICML 2021*.

[HNN] Greydanus, S. et al. (2019). Hamiltonian neural networks. *NeurIPS 2019*.

[Latent-ODE] Rubanova, Y. et al. (2019). Latent ordinary differential equations for irregularly-sampled time series. *NeurIPS 2019*.

[LNN] Cranmer, M., Greydanus, S., & Kochkov, D. (2020). Lagrangian neural networks. *ICLR 2020 workshop*.

[Neural-SDE] Li, X. et al. (2020). Scalable gradients for stochastic differential equations. *AISTATS 2020*.

[Störmer-Verlet] Leimkuhler, B., & Reich, S. (2004). *Simulating Hamiltonian Dynamics*. Cambridge University Press.

Meta-Learning e Adattamento Rapido

[MAML] Finn, C., Abbeel, P., & Levine, S. (2017). Model-agnostic meta-learning for fast adaptation of deep networks. *ICML 2017*.

[Meta-World-Model] Clavera, I. et al. (2018). Model-based meta-learning for fast adaptation. *CORL 2018*.

[Reptile] Nichol, A., & Schulman, J. (2018). On first-order meta-learning algorithms. *arXiv:1803.02999*.

[VariBAD] Zintgraf, L. et al. (2020). VariBAD: A very good method for Bayes-adaptive deep RL via meta-learning. *ICLR 2020*.

[Von-Oswald-2023] Von Oswald, J. et al. (2023). Transformers learn in-context by gradient descent. *ICML 2023*.

Causalità, Rappresentazioni Strutturali e Grounding

[Barlow-IRM] Arjovsky, M. et al. (2019). Invariant risk minimization. *arXiv:1907.02893*.

[Causal-Rep-Learning] Schölkopf, B. et al. (2021). Toward causal representation learning. *Proceedings of the IEEE*, 109(5), 612–634.

[CLEVRER] Yi, K. et al. (2020). CLEVRER: Collision events for video representation and reasoning. *ICLR 2020*.

[ICP] Peters, J., Bühlmann, P., & Meinshausen, N. (2016). Causal inference by using invariant prediction. *Journal of the Royal Statistical Society B*, 78(5), 947–1012.

[Locatello-Disentanglement] Locatello, F. et al. (2019). Challenging common assumptions in the unsupervised learning of disentangled representations. *ICML 2019*.

[NOTEARS] Zheng, X. et al. (2018). DAGs with NO TEARS: Continuous optimization for structure learning. *NeurIPS 2018*.

Leggi di Scala e Modelli Fondazionali

[Chinchilla] Hoffmann, J. et al. (2022). Training compute-optimal large language models. *NeurIPS 2022*.

[Emergent-Abilities] Wei, J. et al. (2022). Emergent abilities of large language models. *TMLR 2022*.

[GPT-4] OpenAI (2023). GPT-4 technical report. *arXiv:2303.08774*.

[Kaplan-Scaling] Kaplan, J. et al. (2020). Scaling laws for neural language models. *arXiv:2001.08361*.

[Sora-OpenAI] OpenAI (2024). Video generation models as world simulators. *Technical Report*. openai.com/research/video-generation-models-as-world-simulators.

[UniSim] Yang, M. et al. (2023). UniSim: Learning interactive real-world simulators. *ICLR 2024*.

Geometria Differenziale, Information Geometry e Ottimizzazione

[Amari-1998] Amari, S.-I. (1998). Natural gradient works efficiently in learning. *Neural Computation*, 10(2), 251–276.

[Amari-Nagaoka-2000] Amari, S.-I., & Nagaoka, H. (2000). *Methods of Information Geometry*. American Mathematical Society.

[Foret-SAM] Foret, P. et al. (2021). Sharpness-aware minimization for efficiently improving generalization. *ICLR 2021*.

[Hyperbolic-NN] Ganea, O. et al. (2018). Hyperbolic neural networks. *NeurIPS 2018*.

[Jacot-NTK] Jacot, A., Gabriel, F., & Hongler, C. (2018). Neural tangent kernel. *NeurIPS 2018*.

[KFAC] Martens, J., & Grosse, R. (2015). Optimizing neural networks with Kronecker-factored approximate curvature. *ICML 2015*.

[Kingma-Adam] Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. *ICLR 2015*.

[Nickel-Kiela] Nickel, M., & Kiela, D. (2017). Poincaré embeddings for learning hierarchical representations. *NeurIPS 2017*.

[Peebles-DiT] Peebles, W., & Xie, S. (2023). Scalable diffusion models with transformers. *ICCV 2023*.

Modelli Generativi Avanzati

[Consistency-Models] Song, Y. et al. (2023). Consistency models. *ICML 2023*.

[DDPM] Ho, J. et al. (2020). Denoising diffusion probabilistic models. *NeurIPS 2020*.

[DDIM] Song, J., Meng, C., & Ermon, S. (2021). Denoising diffusion implicit models. *ICLR 2021*.

[Flow-Matching] Lipman, Y. et al. (2022). Flow matching for generative modeling. *ICLR 2023*.

[Glow] Kingma, D. P., & Dhariwal, P. (2018). Glow: Generative flow with invertible 1×1 convolutions. *NeurIPS 2018*.

[IDDPM] Nichol, A., & Dhariwal, P. (2021). Improved denoising diffusion probabilistic models. *ICML 2021*.

[Latent-Diffusion] Rombach, R. et al. (2022). High-resolution image synthesis with latent diffusion models. *CVPR 2022*.

[Score-SDE] Song, Y. et al. (2021). Score-based generative modeling through stochastic differential equations. *ICLR 2021*.

[Schrödinger-Bridge] De Bortoli, V. et al. (2021). Diffusion Schrödinger bridge. *NeurIPS 2021*.

Apprendimento Multimodale e Grounding Visio-Linguistico

[CLIP] Radford, A. et al. (2021). Learning transferable visual models from natural language supervision. *ICML 2021*.

[CLIPort] Shridhar, M. et al. (2022). CLIPort: What and where pathways for robotic manipulation. *CoRL 2022*.

[CoCa] Yu, J. et al. (2022). CoCa: Contrastive captioners are image-text foundation models. *TMLR 2022*.

[DALL-E-2] Ramesh, A. et al. (2022). Hierarchical text-conditional image generation with CLIP latents. *arXiv:2204.06125*.

[FLAVA] Singh, A. et al. (2022). FLAVA: A foundational language and vision alignment model. *CVPR 2022*.

[Modality-Gap] Liang, V. W. et al. (2022). Mind the gap: Understanding the modality gap in multi-modal contrastive representation learning. *NeurIPS 2022*.

[SigLIP] Zhai, X. et al. (2023). Sigmoid loss for language image pre-training. *ICCV 2023*.

POMDP, Inferenza Attiva e Pianificazione sotto Incertezza

[Active-Inference] Friston, K. et al. (2017). Active inference: A process theory. *Neural Computation*, 29(1), 1–49.

[Deep-POMDP] Igl, M. et al. (2018). Deep variational reinforcement learning for POMDPs. *ICML 2018*.

[MERLIN] Wayne, G. et al. (2018). Unsupervised predictive memory in a goal-directed agent. *arXiv:1803.10760*.

[PETS] Chua, K. et al. (2018). Deep reinforcement learning in a handful of trials. *NeurIPS 2018*.

[RND] Burda, Y. et al. (2019). Exploration by random network distillation. *ICLR 2019*.

[SARSOP] Kurniawati, H., Hsu, D., & Lee, W. S. (2008). SARSOP: Efficient point-based POMDP planning. *RSS 2008*.

Memoria, Architetture Episodiche e Apprendimento Continuo

[Compressive-Transformer] Rae, J. W. et al. (2020). Compressive transformers for long-range sequence modelling. *ICLR 2020*.

[DNC] Graves, A. et al. (2016). Hybrid computing using a neural network with dynamic external memory. *Nature*, 538, 471–476.

[EWC] Kirkpatrick, J. et al. (2017). Overcoming catastrophic forgetting in neural networks. *PNAS*, 114(13), 3521–3526.

[iCaRL] Rebuffi, S.-A. et al. (2017). iCaRL: Incremental classifier and representation learning. *CVPR 2017*.

[NEC] Pritzel, A. et al. (2017). Neural episodic control. *ICML 2017*.

[NeRF] Mildenhall, B. et al. (2020). NeRF: Representing scenes as neural radiance fields. *ECCV 2020*.

[NTM] Graves, A. et al. (2014). Neural Turing machines. *arXiv:1410.5401*.

[SAE-Anthropic] Bricken, T. et al. (2023). Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*.

Fisica Statistica, Teoria Matrici Casuali e Analisi Topologica

[Belkin-Double-Descent] Belkin, M. et al. (2019). Reconciling modern machine learning practice and the bias-variance trade-off. *PNAS*, 116(32), 15849–15854.

[Carlsson-2009] Carlsson, G. (2009). Topology and data. *Bulletin of the AMS*, 46(2), 255–308.

[Chentsov-1982] Chentsov, N. N. (1982). *Statistical Decision Rules and Optimal Inference*. AMS.

[Cohen-Steiner-Stability] Cohen-Steiner, D. et al. (2007). Stability of persistence diagrams. *Discrete & Computational Geometry*, 37(1), 103–120.

[Edge-of-Stability] Cohen, J. M. et al. (2021). Gradient descent on neural networks typically occurs at the edge of stability. *ICLR 2021*.

[Edelsbrunner-PH] Edelsbrunner, H. et al. (2000). Topological persistence and simplification. *FOCS 2000*.

[Heavy-Tails-Martin] Martin, C. H., & Mahoney, M. W. (2021). Implicit self-regularization in deep neural networks. *JMLR*, 22(116), 1–73.

[Mapper] Singh, G., Mémoli, F., & Carlsson, G. (2007). Topological methods for the analysis of high-dimensional data sets and 3D object recognition. *Symposium on Point-Based Graphics*.

[Parisi-RSB] Parisi, G. (1980). The order parameter for spin glasses: A function on the interval [0,1]. *Journal of Physics A*, 13(3), 1101.

Modelli di Linguaggio come World Model e RLHF

[Chain-of-Thought] Wei, J. et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *NeurIPS 2022*.

[DPO] Rafailov, R. et al. (2023). Direct preference optimization. *NeurIPS 2023*.

[Generative-Agents] Park, J. S. et al. (2023). Generative agents: Interactive simulants of human behavior. *UIST 2023*.

[InstructGPT] Ouyang, L. et al. (2022). Training language models to follow instructions. *NeurIPS 2022*.

[Othello-GPT] Li, K. et al. (2023). Emergent world representations: Exploring a sequence model trained on a synthetic task. *ICLR 2023*.

[RAG] Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *NeurIPS 2020*.

[ROME] Meng, K. et al. (2022). Locating and editing factual associations in GPT. *NeurIPS 2022*.

[Tree-of-Thoughts] Yao, S. et al. (2023). Tree of thoughts: Deliberate problem solving with large language models. *NeurIPS 2023*.